

```

1  ; Multi-IO
2  ; (c)W.Römer 88-2017
3  ;
4  ; Bearbeitung: 05.11.2017      RTC OK
5  ;              02.11.2017      RTC Schreibroutine ueber Buffer
6  ;              01.11.2017      RTC-Routinen ueberarbeitet
7  ;              02.05.2017      RTC-Test
8  ;              01.05.2017      RTC-Schaltung berichtigt
9  ;              26.04.2017      Uhr schreiben/lesen
10 ;              08.03.2017      Centronics
11 ;              05.03.2017      com1/2 OK
12 ;              03.03.2017      IO-byte OK
13 ;              02.03.2017      Hardware neu aufgebaut
14 ;              08.02.2017      com1,com2 nicht stabil ?
15 ;              06.02.2017      Softwaregrundfunktionen
16 ;              IO-Byte OK
17 ;              Modusumschaltung OK
18 ;              Centronics OK
19 ;              parallele Ausgabe OK
20 ;              xx.04.1988      Hardware erstellt
21 ;
22 ; 2xV24
23 ; 1xCentronics
24 ; 1xparallele Ein/Ausgabe
25 ;
26 ; Baudrate wird mit Timer 8254 erzeugt
27 ; - com1 300Bd - 9600Bd TX/RX
28 ; - com2 300Bd - 9600Bd TX/RX
29 ;
30 ; IO-Byte fuer die Konfiguration
31 ;
32 ;-----
33 ;
34 ; PIO1 Centronics und UNI-IO Kanal
35 ;
36 pio1a    equ    03h    ; Centronic's
37 pio1b    equ    13h    ; frei
38 pio1c    equ    23h    ; Steuerung Centronics
39          ;          Bit 7 6 5 4 3 2 1 0
40          ;          I I I I I I + + frei
41          ;          I I I I I I
42          ;          I I I I I + - - /Error      <--
43          ;          I I I I + - - - INTRa       -->
44          ;          I I I + - - - - Papierende  <--
45          ;          I I + - - - - - BUSY        <--
46          ;          I + - - - - - - /ACK        <--
47          ;          + - - - - - - - /STROBE     -->
48
49 pio1s    equ    33h    ; Steuerregister
50 bap1     equ    0abh    ; 1010 1011
51          ; Kanal A Modus 1 Ausgabe, Kanal B Modus 0 Eingabe
52          ; Kanal C Eingabe
53 ;
54 ; PIO2 IO-Byte und Systemuhr
55 ;
56 pio2a    equ    43h    ;
57          ; RTC 58321 Daten
58          ;          Bit 7 6 5 4 3 2 1 0
59          ;          x x x x + + + + Daten 0-3  <->
60          ; Bit 4-7 frei
61          ;
62 pio2b    equ    53h    ; IO-byte DIP-Schalter      <--
63          ;
64 pio2c    equ    63h    ; RTC 58321 Control
65          ;          Bit 7 6 5 4 3 2 1 0
66          ;          x x x I I I I I
67          ;          x x x I I I I + WR          -->
68          ;          x x x I I I + - RD          -->
69          ;          x x x I I + - - AD/WR       -->
70          ;          x x x I + - - - CS          -->
71          ;          x x x + - - - - Stop        -->

```

```

72      ; Bit 5-7 frei                                -->
73      ;
74      pio2s    equ    73h    ; Steuerregister
75      ; Betriebsarten
76      bap2     equ    82h    ; 1000 0010
77      ; Kanal A Modus 0 Ausgabe
78      ; Kanal B Modus 0 Eingabe
79      ; Kanal C lo Ausgabe
80      ; Kanal C hi Ausgabe
81      ;
82      bap2rdu  equ    92h    ; 1001 0010
83      ; Kanal A Modus 0 Eingabe
84      ; Kanal B Modus 0 Eingabe
85      ; Kanal C lo Ausgabe
86      ; Kanal C hi Ausgabe
87      ;
88      ; -----
89      ; Baudraten
90      ; Wert als BCD, da Timer auf BCD Zaehler eingestellt wird
91      ;
92      ; fuer 6,144Mhz
93      ;
94      ; 9600Bd = 20h lowByte
95      ; 4800Bd = 40h .
96      ; 2400Bd = 80h .
97      ; 1200Bd = 60h lowByte
98      ;          01h highByte
99      ; -----
100
101      tim0     equ    83h    ; BD-Rate fuer com1
102      tim1     equ    93h    ; BD-Rate fuer com2
103      tim2     equ    0a3h   ; frei
104      tims     equ    0b3h   ; Steuerregister
105      ; Timer0-2 als BCD-Zaehler, Rechteckgenerator,
106      ; erst Timewert-Low, dann Timewert-High laden
107      bati0    equ    37h    ; 0011 0111
108      bati1    equ    77h    ; 0111 0111
109      bati2    equ    0b7h   ; 1011 0111
110      ;
111      com1s    equ    0c3h   ; Steuerregister
112      com1d    equ    0d3h   ; Daten com1
113      bacom1   equ    15h    ; Betriebsart (8bit,1Stop,ohne Paritaet,
Takt/16) ;
114      ;
115      com2s    equ    0e3h   ; Steuerregister
116      com2d    equ    0f3h   ; Daten com2
117      bacom2   equ    15h    ; Betriebsart (8bit,1Stop,ohne Paritaet, Takt/16)
118      ;
119      ;
120      ; -----
121      ;
122      ; Vorspann Parameterbereich
123      ;
124      ; Monitorprogramme
125      ;
126      CI       EQU    0043h   ; CI    Tastaturabfrage wartet auf Taste ASCII(A)
127      CO       EQU    0049h   ; CO    Bildschirmausgabe ASCII(C) Ziel (UPDATE)
128      LOM      EQU    004Fh   ; LO    Drucker Ueber RS232
129      CSTS     EQU    0052h   ; CSTS  Tastaturstatus Taste Z=0, k.Taste Z=1
130      FILL     EQU    0085h   ; FILL  Speicher mit Konstante fuellen
SRCBEG, SRCEND, DSTBEG
131      COPYM    EQU    0088h   ; COPY  Speicherbereich verschieben
132      onesec   EQU    001bh   ; Wartezeit 1Sekunde
133      dely1    EQU    000bh   ;          1ms
134      ;
135      BELPOR    EQU    52h    ; Port fuer Summer
136      BYTFF    EQU    1Ah    ; PORT FF auf Singel-Step-Modul
137      COMREG    EQU    48h    ; Register Adressen
138      TRKREG    EQU    49h    ; WD1770
139      SECREG    EQU    4Ah    ;
140      DATREG    EQU    4BH    ;

```

```

141 STATUS EQU COMREG
142 CTRL EQU 40H
143 ;
144 MOVID EQU 1003h ; Warmstart Videomonitor
145 BREAK EQU 100Fh ; Unterbrechung mit CTRL-C
146 PARIN EQU 1012h ;
147 STRING EQU 1021h ; STRING
148 ERROR EQU 1027h ; Error Videomonitor
149 CLRVID EQU 102Dh ; CLRVID
150 CLRLIN EQU 1030h ; CLRLIN
151 CRSPOS EQU 1033h ; CRSPOS
152 CRSON EQU 1036h ; CRSON
153 CRSOFF EQU 1039h ; CRSOFF
154 BYTOT EQU 103Fh ; BYTOT (A) zweistellig darstellen Adresse (HL)
155 ADROT EQU 103ch ; (DE) vierstellig ausgeben
156 CRLF EQU 137bh ; gibt CR/LF aus
157 ;
158 ; Exteren Programme
159 ;
160 FDC EQU 2000h ; Floppy
161 EDIT EQU 7000h ; Editor
162 ASS: EQU 6000h ; Assembler
163 DISSK EQU 6C00h ; Disassembler
164 ;
165 STACK EQU 2F00H ;
166 TRAP EQU 2F8Ch
167 RST55 EQU 2F98h
168 RST551 EQU 2F99h
169 ;
170 HORDSP EQU 2E50h ; 40/80 Zeichen pro Zeile
171 BEGIN EQU 2E54h ; Anfangsadresse letzte Zeile
172 UPDATE EQU 2E56h ; aktuelle Cursorposition
173 ;
174 SRCBEG EQU 2FBFh ; Startadresse
175 SRCEND EQU 2FC1h ; Endadresse
176 DSTBEG EQU 2FC3h ; Anfang Zieladresse/Parameter
177 DSTEND EQU 2FC5h ;
178 LINZ EQU 2E54h
179 ;
180 ;uhrbuf: equ 9800h ; Uhr-BCD
181 ;ubfasc: equ 982fh ; Uhr-ASCII Ende des Buffers
182 ;
183 ;uhrbuf: equ 2fdch ; Uhr-BCD
184 ;ubfasc: equ 2feah ; Uhr-ASCII
185 ;
186 VIDBEG EQU 3000h ; Bildschirmadresse oben links
187 VIDEND EQU 377Fh ; unten rechts
188 PRTFLG EQU 37AFh ;
189 RAMBEG EQU 37B0h ; Anfang Parameterbereich Videomonitor
190 ;
191 ; Buffer Adressen
192 ;
193 ;
194 BATBUF EQU 2D00h
195 FDCBUF EQU 2F70h
196 CPYBUF EQU 8000h
197 ;
198 ORG FDCBUF
199 ;
200 TRKNR: DS 1h ; Spurnummer
201 SECNR: DS 1h ; Sektornummer
202 SIDNR: DS 1h ; Seite 0/1
203 SELBYT: DS 1h ; SelectByte Bit1,2 = Laufwerksnummer
204 SELOUT: DS 1h
205 TRACK0: DS 4h
206 RWBEF: DS 1h
207 DATADR: DS 2h ; Buffer-Adresse r/w Sektoren
208 INTPRG: DS 2h
209 RWSUB: DS 1h
210 RWSUB1: DS 2h
211 HIBUF: DS 2h

```

```

212 BUFDST: DS 1h
213 BUFSRC: DS 1h ; IM CF-Betrieb = Anzahl der Sektoren die ¼bertragen werden
214 CPYEND: DS 1h
215 ;
216 TIMEOUT EQU 32h ;
217 FQUARZ EQU 06h ;
218 DLY100 EQU 13h ;
219 SECZAL EQU 16 ;
220 SECLDD EQU 1 ; 256 Bytes
221 ;INTON EQU 1Eh ; RST 5.5 freigeben
222 ;INTOF EQU 1Fh ; RST 5.5 sperren
223 ;
224 ;
225 ;-----
226 ;
227 ; Testroutinen, nach Reset muss immer der "Kaltstart"
228 ; aufgerufen werden damit alle Schnittstellen initialisiert werden.
229 ; Die Routinen werden mit beliebigem Tastendruck beendet.
230 ;
231
232 org 9000h ; Testadresse
233 ;
234 kstart: jmp kalt ; INI
235 jmp port ; Portadresse ansprechen
236 jmp rdio ; IO-Byte lesen
237 jmp para ; Test UNI Parallele Aus/Eingabe
238 jmp print ; Printkanal
239 jmp comtx ; Byte senden an comx
240 jmp comrx ; Byte lesen con comx
241 jmp ende ; Dummy
242 jmp uhrdr ; RTC lesen
243 jmp uhrstart ; Uhrstellen
244 jmp uhrasci ; Uhrzeit in ASCII
245 jmp uhrstrg ; Uhrzeit ausgeben
246 jmp 0000h ;
247
248 ende: lxi h,meld10 ;
249 call string ;
250 jmp movid ; zurueck zum Monitor
251 ;
252 ;-----
253 ;
254 kalt: call mioini ; IO-Bausteine initialisieren
255 call fillb ; Buffer 8000h mit Ascii fuellen
256 lxi h,meld1 ;
257 call string ;
258 jmp rdio ; weiter mit IO-byte ausgeben
259 ;
260 ;-----
261 ;
262 ;
263 mioini: mvi a,bap1 ; PIO-Betriebsarten einstellen
264 out pio1s ;
265 mvi a,bap2 ;
266 out pio2s ;
267 xra a ;
268 out pio2b ;
269 out pio2c ;
270 ;
271 mvi a,bacom1 ; com1-Betriebsart
272 out com1s ;
273 mvi a,bacom2 ; com2-Betriebsart
274 out com2s ;
275 ;
276 mvi a,bati0 ; Timer-Betriebsarten
277 out tims ;
278 mvi a,bati1 ;
279 out tims ;
280 mvi a,bati2 ;
281 out tims ;
282 ;

```

```

283      lxi      d,0020h      ; 9600Bd
284      mov      a,e          ; in Timer 0 laden
285      out      tim0         ;
286      mov      a,d          ;
287      out      tim0         ;
288      ;
289      mov      a,e          ; in Timer 1 laden
290      out      tim1         ;
291      mov      a,d          ;
292      out      tim1         ;
293      ;
294      mov      a,e          ; in Timer 2 laden
295      out      tim2         ;
296      mov      a,d          ;
297      out      tim2         ;
298      ;
299      mvi      a,03h        ; Reset UART1/2
300      out      com1s        ;
301      out      com2s        ;
302      ret
303      ;
304      ;
305      ;-----
306      ;
307 port:   lxi      h,meld6    ;
308         call    string      ;
309         mvi     a,tim0      ; Portadresse ausgeben
310         call    bytot
311         mvi     b,00        ;
312 port1:  inr      b          ;
313         mov     a,b         ;
314         out     tim0        ;
315         call    csts        ;
316         jz      port1       ;
317         jmp     ende
318      ;
319      ;-----
320      ;
321 rdio:   lxi      h,meld3    ; Meldung
322         call    string      ;
323 rloop:  in       pio2b      ; Befehl
324         call    bytot      ; IO-Byte ausgeben
325         mvi     c,0dh       ; "CR"
326         call    co         ; ausgeben
327         call    csts        ;
328         jz      rloop       ; Schleife bis eine Taste betaetigt wird
329         jmp     ende        ;
330      ;
331      ;-----
332      ;
333      ; fuer com1/2 entsprechenden Port eintragen
334      ;
335 comtx:  lxi      h,meld1    ;
336         call    string      ;
337
338         call    mioini      ;
339         lxi     h,meld2
340         call    string
341         ;
342 comt1:  mvi      c,41h      ; "A"
343         call    com2t       ; <-- Port eintragen
344         call    csts        ;
345         jz      comt1       ; Schleife bis eine Taste betaetigt wird
346         jmp     ende        ;
347      ;
348      ;-----
349      ;
350 comrx:  lxi      h,meld7    ;
351         call    string      ;
352
353         call    mioini      ;

```

```

354      lxi      h,meld2
355      call     string
356      ;
357      comrl    call     com2r      ; Daten empfangen
358      mov      c,a                ; <-- Port eintragen
359      call     co
360      call     csts
361      jz       comrl              ; Schleife bis eine Taste betaetigt wird
362      jmp      ende
363      ;
364      ;-----
365      ;
366      para:    lxi      h,meld4      ;
367      call     string
368      mvi      b,00
369      mvi      a,bap2
370      out      pio2s
371      paral:   in       pio1a        ; IO-Byte
372      out      pio2b        ; auf Parallel ausgeben
373      call     csts
374      jz       paral
375      jmp      ende
376      ;
377      ;-----
378      ;
379      ; Druckt 64 ASCII Zeichen aus Speicher 8000h
380      ;
381      print:   lxi      h,meld5      ;
382      call     string
383      mvi      a,bapl
384      out      piols
385      lxi      h,8000h      ; Quelle ASCII
386      mvi      b,40
387      prt1:    in       piolc
388      ani      20h          ; BUSY maskieren
389      jnz      prt1
390      mov      a,m          ; Daten holen
391      out      pio1a        ; an Drucker
392      dcr      b
393      jz       prtend       ; alle Zeichen
394      inc      h
395      jmp      prt1
396
397      prtend:  jmp      ende
398      ;
399      ; Buffer mit ASCII fuellen
400      fillb:   lxi      h,8000h      ; Startadresse Buffer
401      mvi      b,40h
402      mvi      c,40h
403      filbl:   mov      m,c
404      inc      h
405      inc      c
406      dcr      b
407      jnz      filbl
408      xra      a
409      mov      m,a          ; Ende String 00h
410      ret
411      ;
412      ;-----
413      ;
414      ; Byte v24 senden (C)
415      ; wartet bis Empfaenger bereit ist
416      ;
417      ;
418      ;
419      com1t:   mvi      a,55h      ;
420      out      com1s          ; clk/16,8bit,1stop,ohne paritaet , RTS on
421      com1t_l: in       com1s      ; Warten bis Senderegister leer
422      ani      02H
423      jz       com1t_l
424      mov      a,c

```

```

425         out      com1d          ; Daten senden
426         ret
427     ;
428 com2t: mvi      a,55h          ;
429         out      com2s          ; clk/16,8bit,1stop,ohne paritaet , RTS on
430 com2t_l: in      com2s          ; Warten bis Senderegister leer
431         ani      02H          ;
432         jz       com2t_l       ;
433         mov      a,c          ;
434         out      com2d          ; daten senden
435         ret
436     ;-----
437     ;
438     ;
439     ; Byte von V24 empfangen (A)
440     ; wartet bis Zeichen vorhanden ist
441     ;
442 com1r: mvi      a,15h          ;
443         out      com1s          ; clk/16,8bit,1stop,ohne paritaet , RTS off
444 com1r_l: in      com1s          ; Warten bis Empfangsregister voll
445         ani      01H          ;
446         jz       com1r_l       ; warten
447         mvi      a,55h          ; RTS on
448         out      com1s          ;
449         in       com1d          ; Daten auslesen
450         ret
451     ;
452 com2r: mvi      a,15h          ;
453         out      com2s          ; clk/16,8bit,1stop,ohne paritaet , RTS off
454 com2r_l: in      com2s          ; Warten bis Empfangsregister voll
455         ani      01H          ;
456         jz       com2r_l       ; warten
457         mvi      a,55h          ; RTS on
458         out      com2s          ;
459         in       com2d          ; Daten auslesen
460         ret
461     ;
462     ;=====
463     ;
464     ; Echzeituhr mit RTC-58321
465     ;
466     ; Betriebsarten PIO-2:
467     ; bap2          = 82h      1000 0010 (nach Reset)
468     ;                                     Kanal A Modus 0 Ausgabe
469     ;                                     Kanal B Modus 0 Eingabe
470     ;                                     Kanal C Ausgabe
471     ; bap2rdu       = 92h      1001 0010 (Uhr lesen)
472     ;                                     Kanal A Modus 0 Eingabe
473     ;                                     Kanal B Modus 0 Eingabe
474     ;                                     Kanal C Ausgabe
475     ;
476     ; Daten/Adressen ueber Port pio2a = 43h
477     ;      Bit 7 6 5 4 3 2 1 0
478     ;      x x x x + + + + Daten 0-3
479     ;
480     ;      Bit 4-7 frei
481     ;
482     ; Steuerung ueber Port pio2c = 63h
483     ;
484     ;      Bit 7 6 5 4 3 2 1 0
485     ;      x x x I I I I I
486     ;      x x x I I I I + WR
487     ;      x x x I I I + - RD
488     ;      x x x I I + - - ADR-WR
489     ;      x x x I + - - - CS
490     ;      x x x + - - - - Stop
491     ;
492     ; RTC Registeradressen
493     ; Adressen      Bedeutung      Gueltigkeit Datenbit 0-3
494     ;
495     ;      3 2 1 0

```

```

496 ; 0      0 0 0 0 = Sekunden-Einer      0 - 9
497 ; 1      0 0 0 1 = Sekunden-Zehner     0 - 5
498 ; 2      0 0 1 0 = Minuten-Einer       0 - 9
499 ; 3      0 0 1 1 = Minuten-Zehner      0 - 5
500 ; 4      0 1 0 0 = Stunden-Einer        0 - 9
501 ; 5      0 1 0 1 = Stunden-Zehner      0 - 1 / 0 - 2 im 24Std-Modus
502 ; 6      0 1 1 0 = Wochentag           0 - 6
503 ; 7      0 1 1 1 = Tag-Einer           0 - 9
504 ; 8      1 0 0 0 = Tag-Zehner          0 - 3 (D2 = 1 fuer Februar mit 29Tagen/sonst
      28Tage)
505 ; 9      1 0 0 1 = Monat-Einer         0 - 9
506 ; A      1 0 1 0 = Monat-Zehner        0 - 1
507 ; B      1 0 1 1 = Jahr-Einer          0 - 9
508 ; C      1 1 0 0 = Jahr-Zehner         0 - 9
509 ;
510 ;
511 ; Register:      A,DE,HL
512 ;                (B) : CTL-Maske fuer Port 2c
513 ;                (C) : Adresse
514 ;
515 ; Portzuweisung (neus Label fuer die Systemuhrabfrage)
516 ;
517 rtc_ctrl equ      pio2c
518 rtc_dat  equ      pio2a
519 ;
520 ;                ; 7654 3210
521 ;                ;
522 rtccs     equ      18h                ; 0001 1000 = RTC Selektieren und Zaehler-Stop
523 rtcwr     equ      19h                ; 0001 1001 = RTC Datennibble schreiben
524 rtcrd     equ      1ah                ; 0001 1010 = RTC Datennibble lesen
525 rtcawr    equ      1ch                ; 0001 1100 = RTC Registeradresse schreiben
526 ;
527 ;
528 ; Verzoeegerungsroutine
529 ;
530 dely10: ret                        ; ca 10µs
531 ;
532 ;
533 ;-----
534 ;
535 ; RTC-58321 mit Daten laden
536 ; diese muessen im uhrbuf vorher bereitgestellt werden
537 ; siehe Registertabelle
538 ;
539 uhrstart: xra      a                  ;
540           lxi      h,meld8            ;
541           call     string              ;
542 ;
543           lxi      h,uhrbuf           ;
544           mvi      c,00h              ; Zeiger fuer RTC-Register
545           mvi      b,0dh              ; Zaheler
546 ;
547           mvi      a,bap2             ; Ausgabe
548           out      pio2s              ;
549 ;
550 uhr_sl: mov      a,m                  ; bit holen
551           call     wr_rtc              ;
552           inx      h                  ; naechstes byte
553           dcr      b                  ;
554           jnz      uhr_sl             ;
555           xra      a                  ;
556           out      rtc_ctrl           ; RTC Steuereingaenge auf 0
557 ;
558           jmp      ende               ;
559 ;
560 ;
561 ; wr_rtc Daten in RTC-58321 schreiben
562 ;
563 wr_rtc: push     psw                  ; daten retten
564           mov      a,c                ; Registeradresse
565           out      rtc_dat            ;

```



```

566      call    dely10                ;
567      mvi     a,rtcawr              ; Adress-Schreibimpuls
568      out     rtc_ctrl              ;
569      call    dely10                ;
570      mvi     a,rtccs               ;
571      out     rtc_ctrl              ;
572
573      call    dely10                ;
574      pop     psw                   ;
575      out     rtc_dat               ; Wert in RTC laden
576      mvi     a,rtcwr              ; Schreibimpuls
577      out     rtc_ctrl              ;
578      call    dely10                ;
579      mvi     a,rtccs               ;
580      out     rtc_ctrl              ;
581
582      inr     c                     ; naechste Adresse einstellen
583      ret
584
585
586      ;-----
587      ;
588      ; Uhrzeit aus RTC-58321 lesen
589      ;
590      ; (c) Registeradresse
591      ; (hl) Bufferadresse
592      ;
593 uhrd:  lxi     h,meld9              ;
594      call    string                ;
595      call    uhr_rd                ;
596      jmp     ende                  ;
597
598 uhr_rd: lxi     h,uhrbuf             ;
599      mvi     c,00h                 ; Zeiger auf sek-einer
600      mvi     a,bap2                ; Ausgabe
601      out     pio2s                 ;
602      mvi     a,rtccs               ; RTC stop/cs
603      out     rtc_ctrl              ;
604 urdloop: call    dely10             ;
605      mov     a,c                   ;Adresse einstellen
606      out     rtc_dat               ;
607
608      call    dely10                ;
609      mvi     a,rtcawr              ; Adresseschreiben
610      out     rtc_ctrl              ; ADR-Schreibimpuls
611      call    dely10                ;
612      mvi     a,rtccs               ;
613      out     rtc_ctrl              ;
614
615      mvi     a,bap2rdu             ; Kanal auf Eingabe umschalten
616      out     pio2s                 ;
617
618      call    dely10                ; Register lesen
619      mvi     a,rtcrd               ;
620      out     rtc_ctrl              ;
621      call    dely10                ;
622      in      rtc_dat               ;
623      ani     0fh                   ;oberes nibble ausblenden
624      mov     m,a                   ;
625      mvi     a,rtccs               ;
626      out     rtc_ctrl              ; Leseimpuls beenden
627
628      mvi     a,bap2                ; Ausgabe
629      out     pio2s                 ;
630
631      inx     h                     ; naechste Adresse einstellen
632      inr     c                     ;
633      mov     a,c                   ; alle Register gelesen
634      cpi     0dh                   ;
635      jnz     urdloop               ; nein
636

```

```

637     uhr_end: mvi     a,00h           ; ende
638             out      rtc_ctrl       ;
639             ret                          ;
640
641
642 ; -----
643 ;
644 ;
645 uhrasci: call     uhr_asc             ; Uhr lesen und in ASCII wandeln
646             lxi      h,uascbuf       ; Startadr. ASCIIBuffer
647             call     string          ; ausgeben
648             jmp      ende
649
650 ; Uhrzeit im ASCII-Buffer schreiben
651 ; ! wird von rechts nach links eingetragen
652 ;
653 uhr_asc: CALL     uhr_rd             ; Uhr auslesen
654
655 uascw:  lxi      h,uascbuf           ; Startadr. ASCII-Buffer
656             lxi      d,15h           ; Versatz
657             dad      d               ; HL auf ende ASCII-Buffer
658             lxi      d,uhrbuf        ;
659             MVI M,00H                ; 00h eintragen fuer String-Ende
660             DCX H
661             MVI M,68H                ; ASCII "h" speichern
662             CALL h2bcd               ; Sekunden in ASCII Buffer
663             DCX H
664             MVI M,3AH                ; ASCII ":" (Trennzeichen)
665             CALL h2bcd               ; Minuten in ASCII Buffer
666             DCX H
667             MVI M,3AH                ; ASCII ":"
668             CALL h2bcd               ; Stunden in ASCII Buffer
669             DCX H
670             MVI M,3BH                ; ASCII ";"
671             LXI D,u_ja
672             CALL h2bcd               ; Jahr in ASCII Buffer
673             DCX H
674             MVI M,2EH                ; ASCII "."
675             LXI D,u_mo
676             CALL h2bcd               ; Monat in ASCII Buffer
677             DCX H
678             MVI M,2EH                ; ASCII "."
679             LXI D,u_ta
680             CALL h2bcd               ; Tag in ASCII Buffer
681             DCX H
682             MVI M,2CH                ; ASCII ",",
683             DCX H
684             PUSH H                    ;
685             LXI H,wt_tab             ; Tabelle fuer Wochentag
686             LDA u_wt                 ; Wochentag holen
687             ANI 07H                  ; mit Tabelle umsetzen
688             RLC                      ;
689             MOV E,A                  ;
690             MVI D,00H                ;
691             DAD D                    ;
692             MOV E,M                  ;
693             INX H                    ;
694             MOV D,M                  ;
695             POP H                    ;
696             MOV M,E                  ;
697             DCX H                    ;
698             MOV M,D                  ;
699             RET
700 ;
701 ;
702 uhrstrg: lxi      h,meld92            ;
703             call     string          ;
704
705 ;
706 ; h2bcd = hole 2 BCD-Werte aus Buffer
707 ;

```

```

708 h2bcd:  CALL hbcd                ;
709 ;
710 hbcd:   DCX H                    ;
711         LDAX D
712         ADI 30H                  ; Gueltigkeit pruefen
713         CPI 3AH
714         JNC hbcd_er             ; Fehler
715         CPI 30H
716         JNC hbcde               ; OK Wert zuerueckgeben
717 hbcd_er: MVI A, 2DH             ; Wert unguelteig "-" zurueckgeben
718 ;
719 hbcde:  MOV M, A
720         INX D                    ; naechste Bufferadresse
721         RET
722 ;
723 ; Wochentage ..oMiDiMoDrFaSoS
724 ; 1 = Montag, 7 = Sonntag
725 ; Tage sind in umgekehrter Reihenfolge abgelegt
726 ; weil der ASCII-Buffer von hinten nach vorn
727 ; aufgebaut wird
728 ;
729 wt_tab: DB 52h                  ; "R"
730         DB 45h                  ; "E" - Error
731         DB 6Fh                  ; "O" -
732         DB 4Dh                  ; "M" - Montag
733         DB 69h                  ; "i"
734         DB 44h                  ; "D"
735         DB 69h                  ; "i"
736         DB 4Dh                  ; "M"
737         DB 6Fh                  ; "O"
738         DB 44h                  ; "D"
739         DB 72h                  ; "r"
740         DB 46h                  ; "F"
741         DB 61h                  ; "a"
742         DB 53h                  ; "S"
743         DB 6Fh                  ; "O"
744         DB 53h                  ; "S"
745 ;
746 ;
747 ;
748 ;=====
749 ;
750 ; Texte
751 ;
752 meld1:  DB 0Dh                  ; CR
753         DB 0Ah                  ; LF
754         DB "Multi-IO-Test"
755         DB " INI"
756         DB 0Dh                  ;
757         DB 00h
758 ;
759 meld2:  DB 0Dh                  ; CR
760         DB 0Ah                  ; LF
761         DB "Byte an comx"
762         DB " senden"
763         DB 0Dh                  ;
764         DB 00h
765 ;
766 meld3:  DB 0Dh                  ; CR
767         DB 0Ah                  ; LF
768         DB "IO-Byte"
769         DB " lesen"
770         DB 0Dh                  ;
771         DB 00h
772 ;
773 meld4:  DB 0Dh                  ; CR
774         DB 0Ah                  ; LF
775         DB "UNI-Parallel"
776         DB " Ausgabe"
777         DB 0Dh                  ;
778         DB 00h

```

```

779 ;
780 meld5: DB 0Dh ; CR
781         DB 0Ah ; LF
782         DB "Print"
783         DB " Ausgabe"
784         DB 0Dh ;
785         DB 00h
786 ;
787 meld6: DB 0Dh ; CR
788         DB 0Ah ; LF
789         DB "ADR"
790         DB " Test"
791         DB 0Dh ;
792         DB 00h
793 ;
794 meld7: DB 0Dh ; CR
795         DB 0Ah ; LF
796         DB "Byte von comx"
797         DB " empfangen"
798         DB 0Dh ;
799         DB 00h
800 ;
801 meld71: DB 0Dh ; CR
802          DB 0Ah ; LF
803          DB "Reset RTC"
804          DB 0Dh ;
805          DB 00h
806 ;
807 meld8: DB 0Dh ; CR
808         DB 0Ah ; LF
809         DB "Uhr setzen"
810         DB 0Dh ;
811         DB 00h
812 ;
813 meld9: DB 0Dh ; CR
814         DB 0Ah ; LF
815         DB "Uhr lesen"
816         DB 0Dh ;
817         DB 00h
818 ;
819 meld91: DB 0Dh ; CR
820          DB 0Ah ; LF
821          DB "Uhr ASCII"
822          DB 0Dh ;
823          DB 00h
824 ;
825 meld92: DB 0Dh ; CR
826          DB 0Ah ; LF
827          DB "Uhrzeit : "
828          DB 0Dh ;
829          DB 00h
830 ;
831 meld10: DB 0Dh ; CR
832          DB 0Ah ; LF
833          DB "Ende"
834          DB 0Dh ;
835          DB 00h
836 ;
837 ;
838 ;
839 ;-----
840         org     9800h ; Buffer für Uhrzeit in BCD
841
842 uhrbuf: ds      1h ; Sek-Einer
843         ds      1h ; Sek-Zehner
844         ds      1h ; Min-Einer
845         ds      1h ; Min-Zehner
846         ds      1h ; Std-Einer
847         ds      1h ; Std-Zehner
848 u_wt: ds      1h ; Wochentag 1-7 Montag-Sonntag
849 u_ta: ds      1h ; Tag-Einer

```

```
850          ds      1h          ; Tag-Zehner
851  u_mo:      ds      1h          ; Monat-Einer
852          ds      1h          ; Monat-Zehner
853  u_ja:      ds      1h          ; Jahr-Einer
854          ds      1h          ; Jahr-Zehner
855
856          org      9820h        ; Ende ASCII Buffer
857
858  uascbuf:    ds      16h        ;
859
860
```