

```

1  ; Multi-IO V0.11
2  ; (c)W.Römer 88-2017
3  ;
4  ; Bearbeitung: 19.12.2017      V0.11 mit 4800bd OK
5  ;              19.12.2017      IRQ-Fehler gefunden,
6  ;                               RTS Setzen mit call zerro statt
7  ;                               Jump Zerro aufgerufen = Stackfehler
8  ;              16.12.2017      IRQ-Teil Fehleranalyse
9  ;              15.12.2017      IRQ-Teil Fehleranalyse
10 ;              13.12.2017      V24 mit IRQ
11 ;              05.11.2017      RTC OK
12 ;              02.11.2017      RTC Schreibroutine ueber Buffer
13 ;              01.11.2017      RTC-Routinen ueberarbeitet
14 ;              02.05.2017      RTC-Test
15 ;              01.05.2017      RTC-Schaltung berichtigt
16 ;              26.04.2017      Uhr schreiben/lesen
17 ;              08.03.2017      Centronics
18 ;              05.03.2017      com1/2 OK
19 ;              03.03.2017      IO-byte OK
20 ;              02.03.2017      Hardware neu aufgebaut
21 ;              08.02.2017      com1,com2 nicht stabil ?
22 ;              06.02.2017      Softwaregrundfunktionen
23 ;                               IO-Byte OK
24 ;                               Modusumschaltung OK
25 ;                               Centronics OK
26 ;                               parallele Ausgabe OK
27 ;              xx.04.1988      Hardware erstellt
28 ;
29 ;=====
30 ;
31 ; V24 mit IRQ Routine - Test fuer Datenempfang
32 ;
33 ; Daten werden von Interrupt-Serviceroutine im FIFO abgelegt
34 ; und per Kopieroutine im Zielbuffer gespeichert.
35 ; Wenn der Fuehllstand ca. 64% erreicht, wird RTS gesetzt.
36 ; Damit stellt die ISR sicher, dass die Gegenstelle ihren FIFO
37 ; noch leeren kann.
38 ; Anmerkung!
39 ; Moderne PC's sprudeln noch bis zu 16Bytes
40 ; aus der V24 obwohl RTS gesetzt ist.
41 ; Auch bei abgeschalteten FIFO senden sie noch ein bis zwei Bytes
42 ;
43 ;
44 ; Parameter zur Übersicht
45 ;
46 ; buffer          equ 9000h      ; fuer BIN-Datei
47 ; zbuff          equ 8400h      ; Software FIFO
48 ;
49 ;              v-Schreibzeiger (wrz)
50 ;              I
51 ; 8400h I-----+---...---I 843fh (maxbuf)
52 ;              ^               :
53 ;              I- Lesezeiger (rdz)      842ah +----- RTS Setzen (rtsbuf)
54 ;
55 ; maxbuf          equ 3fh        ; Bufferlaenge
56 ; rtsbuf          equ 2ah        ; Bufferposition fuer RTS
57 ;
58 ; buffz           dw 0000h       ; Schreibzeiger fuer Zielbuffer
59 ; wrz             dw 0000h       ; Schreibzeiger fuer IRQ
60 ; rdz             dw 0000h       ; Lesezeiger
61 ; v24flag         db 00h         ; 1000 0001
62 ;               ; :             ;
63 ;               ; :             ; +--- RTS punkt erreicht
64 ;               ; +----- Buffer voll
65 ;
66 ;=====
67 ;
68 ; 2xV24
69 ; 1xCentronics
70 ; 1xparallele Ein/Ausgabe
71 ;

```

```

72 ; Baudrate wird mit Timer 8254 erzeugt
73 ; - com1 300Bd - 9600Bd TX/RX
74 ; - com2 300Bd - 9600Bd TX/RX
75 ;
76 ; IO-Byte fuer die Konfiguration
77 ;
78 ; -----
79 ; Multi-IO-Schnittstellenkarte
80 ; Portbeschreibung: COM1/COM2
81 ;
82 ; Controlregister UART (aus Datenblatt Motorola M6850)
83 ;
84 ;       7 6 5 4 3 2 1 0
85 ;       : : : : : : : :
86 ;       : : : : : : 0 0 = clk/1
87 ;       : : : : : : 0 1 = clk/16
88 ;       : : : : : : 1 0 = clk/64
89 ;       : : : : : : 1 1 = Reset
90 ;       : : : 0 0 0      = ev 2Stop 7Datenbits
91 ;       : : : 0 0 1      = od
92 ;       : : : 0 1 0      = ev 1Stop      (gerade Paritaet)
93 ;       : : : 0 1 1      = od            (ungerade Paritaet)
94 ;       : : : 1 0 0      = -- 2Stop 8Datenbits
95 ;       : : : 1 0 1      = -- 1Stop
96 ;       : : : 1 1 0      = ev 1 Stop
97 ;       : : : 1 1 1      = od
98 ;       : 0 0            = RTS low      TX-IRQ disable
99 ;       : 0 1            = RTS low      TX-IRQ enable
100 ;       : 1 0            = RTS high     TX-IRQ disable
101 ;       : 1 1            = RTS low      TX-IRQ disable
102 ;       1                = RX-IRQ enable
103 ;
104 ;       0 0 0 0 0 0 1 1 = 03h Baustein Reset
105 ;       0 0 0 1 0 1 0 1 = 15h clk/16, 8,n,1
106 ;       0 1 0 1 0 1 0 1 = 55h ... mit RTS=1
107 ;
108 ;       1 0 0 1 0 1 0 1 = 95h clk/16, 8,n,1      ; RTS=0 RX-IRQ enable
109 ;       1 1 0 1 0 1 0 1 = D5h      ""           ; RTS=1      ""
110 ;
111 ;
112 ; Statusregister UART nur lesen
113 ;
114 ;       7 6 5 4 3 2 1 0
115 ;       : : : : : : : :
116 ;       : : : : : : 1 = Empfangsregister voll
117 ;       : : : : : : 1 = Senderegister leer
118 ;       : : : : : 1   = DCD Eingang
119 ;       : : : : x     = CTS Eingang
120 ;       : : : 1       = Fehlerbits      -Frame
121 ;       : : 1         =                  -Overrun
122 ;       : 1           =                  -Paritaet
123 ;       1             = IRQ-Ausgang     (ungerade Paritaet)
124 ;
125 ; -----
126 ;
127 ; PIO1 Centronics und UNI-IO Kanal
128 ;
129 piola equ 03h ; Centronic's
130 piolb equ 13h ; frei
131 piolc equ 23h ; Steuerung Centronics
132 ;       Bit 7 6 5 4 3 2 1 0
133 ;       I I I I I I + + frei
134 ;       I I I I I I
135 ;       I I I I I + - - /Error      <--
136 ;       I I I I + - - - INTRa      -->
137 ;       I I I + - - - - Papierende <--
138 ;       I I + - - - - - BUSY        <--
139 ;       I + - - - - - - /ACK        <--
140 ;       + - - - - - - - /STROBE    -->
141
142 piols equ 33h ; Steuerregister

```

```

143  bap1      equ      0abh      ; 1010 1011
144                                     ; Kanal A Modus 1 Ausgabe, Kanal B Modus 0 Eingabe
145                                     ; Kanal C Eingabe
146  ;
147  ; PIO2 IO-Byte und Systemuhr
148  ;
149  pio2a     equ      43h      ;
150                                     ; RTC 58321 Daten
151                                     ;      Bit 7 6 5 4 3 2 1 0
152                                     ;      x x x x + + + + Daten 0-3  <-->
153                                     ; Bit 4-7 frei
154  ;
155  pio2b     equ      53h      ; IO-byte DIP-Schalter                      <--
156  ;
157  pio2c     equ      63h      ; RTC 58321 Control
158                                     ;      Bit 7 6 5 4 3 2 1 0
159                                     ;      x x x I I I I I
160                                     ;      x x x I I I I + WR                      -->
161                                     ;      x x x I I I + - RD                      -->
162                                     ;      x x x I I + - - AD/WR                  -->
163                                     ;      x x x I + - - - CS                      -->
164                                     ;      x x x + - - - - Stop                  -->
165                                     ; Bit 5-7 frei                      -->
166  ;
167  pio2s     equ      73h      ; Steuerregister
168                                     ; Betriebsarten
169  bap2      equ      82h      ; 1000 0010
170                                     ; Kanal A Modus 0 Ausgabe
171                                     ; Kanal B Modus 0 Eingabe
172                                     ; Kanal C lo Ausgabe
173                                     ; Kanal C hi Ausgabe
174  ;
175  bap2rdu   equ      92h      ; 1001 0010
176                                     ; Kanal A Modus 0 Eingabe
177                                     ; Kanal B Modus 0 Eingabe
178                                     ; Kanal C lo Ausgabe
179                                     ; Kanal C hi Ausgabe
180  ;
181  ;-----
182  ; Baudraten
183  ; Wert als BCD, da Timer auf BCD Zaehler eingestellt wird
184  ;
185  ; fuer 6,144Mhz
186  ;
187  ; 9600Bd = 20h lowByte
188  ; 4800Bd = 40h .
189  ; 2400Bd = 80h .
190  ; 1200Bd = 60h lowByte
191  ;      01h highByte
192  ;-----
193  ;
194  tim0      equ      83h      ; BD-Rate fuer com1
195  tim1      equ      93h      ; BD-Rate fuer com2
196  tim2      equ      0a3h     ; frei
197  tims      equ      0b3h     ; Steuerregister
198                                     ; Timer0-2 als BCD-Zaehler, Rechteckgenerator,
199                                     ; erst Timewert-Low, dann Timewert-High laden
200  bati0     equ      37h      ; 0011 0111
201  bati1     equ      77h      ; 0111 0111
202  bati2     equ      0b7h     ; 1011 0111
203  ;
204  com1s     equ      0c3h     ; Steuerregister
205  com1d     equ      0d3h     ; Daten com1
206  bacom1    equ      15h      ; Betriebsart (8bit,1Stop,ohne Paritaet,
    Takt/16)
207  ;
208  com2s     equ      0e3h     ; Steuerregister
209  com2d     equ      0f3h     ; Daten com2
210  bacom2    equ      15h      ; Betriebsart (8bit,1Stop,ohne Paritaet, Takt/16)
211  ;
212  ;

```

```

213 ;-----
214 ;
215 ; Vorspann Parameterbereich
216 ;
217 ; Monitorprogramme
218 ;
219 CI EQU 0043h ; CI Tastaturabfrage wartet auf Taste ASCII(A)
220 CO EQU 0049h ; CO Bildschirmausgabe ASCII(C) Ziel (UPDATE)
221 LOM EQU 004Fh ; LO Drucker Ueber RS232
222 CSTS EQU 0052h ; CSTS Tastaturstatus Taste Z=0, k.Taste Z=1
223 FILL EQU 0085h ; FILL Speicher mit Konstante fuellen
SRCBEG, SRCEND, DSTBEG
224 COPYM EQU 0088h ; COPY Speicherbereich verschieben
225 onesec EQU 001bh ; Wartezeit 1Sekunde
226 dely1 EQU 000bh ; 1ms
227 ;
228 BELPOR EQU 52h ; Port fuer Summer
229 BYTFF EQU 1Ah ; PORT FF auf Singel-Step-Modul
230 COMREG EQU 48h ; Register Adressen
231 TRKREG EQU 49h ; WD1770
232 SECREG EQU 4Ah
233 DATREG EQU 4BH
234 STATUS EQU COMREG
235 CTRL EQU 40H
236 ;
237 MOVID EQU 1003h ; Warmstart Videomonitor
238 BREAK EQU 100Fh ; Unterbrechung mit CTRL-C
239 PARIN EQU 1012h ;
240 STRING EQU 1021h ; STRING
241 ERROR EQU 1027h ; Error Videomonitor
242 CLRVID EQU 102Dh ; CLRVID
243 CLRLIN EQU 1030h ; CLRLIN
244 CRSPOS EQU 1033h ; CRSPOS
245 CRSON EQU 1036h ; CRSON
246 CRSOFF EQU 1039h ; CRSOFF
247 BYTOT EQU 103Fh ; BYTOT (A) zweistellig darstellen Adresse (HL)
248 ADROT EQU 103ch ; (DE) vierstellig ausgeben
249 CRLF EQU 137bh ; gibt CR/LF aus
250 ;
251 ; Exteren Programme
252 ;
253 FDC EQU 2000h ; Floppy
254 EDIT EQU 7000h ; Editor
255 ASS: EQU 6000h ; Assembler
256 DISSK EQU 6C00h ; Disassembler
257 ;
258 STACK EQU 2F00H ;
259 TRAP EQU 2F8Ch
260 RST55 EQU 2F98h
261 RST551 EQU 2F99h
262 ;
263 HORDSP EQU 2E50h ; 40/80 Zeichen pro Zeile
264 BEGIN EQU 2E54h ; Anfangsadresse letzte Zeile
265 UPDATE EQU 2E56h ; aktuelle Cursorposition
266 ;
267 SRCBEG EQU 2FBFh ; Startadresse
268 SRCEND EQU 2FC1h ; Endadresse
269 DSTBEG EQU 2FC3h ; Anfang Zieladresse/Parameter
270 DSTEND EQU 2FC5h ;
271 LINZ EQU 2E54h
272 ;
273 ;uhrbuf: equ 9800h ; Uhr-BCD
274 ;ubfasc: equ 982fh ; Uhr-ASCII Ende des Buffers
275 ;
276 ;uhrbuf: equ 2fdch ; Uhr-BCD
277 ;ubfasc: equ 2feah ; Uhr-ASCII
278 ;
279 VIDBEG EQU 3000h ; Bildschirmadresse oben links
280 VIDEND EQU 377Fh ; unten rechts
281 PRTFLG EQU 37AFh ;
282 RAMBEG EQU 37B0h ; Anfang Parameterbereich Videomonitor

```

```

283
284 ;
285 ; Buffer Adressen
286 ;
287 BATBUF EQU 2D00h
288 FDCBUF EQU 2F70h
289 CPYBUF EQU 8000h
290 ;
291 ORG FDCBUF
292
293 TRKNR: DS 1h ; Spurnummer
294 SECNR: DS 1h ; Sektornummer
295 SIDNR: DS 1h ; Seite 0/1
296 SELBYT: DS 1h ; SelectByte Bit1,2 = Laufwerksnummer
297 SELOUT: DS 1h
298 TRACK0: DS 4h
299 RWBEF: DS 1h
300 DATADR: DS 2h ; Buffer-Adresse r/w Sektoren
301 INTPRG: DS 2h
302 RWSUB: DS 1h
303 RWSUB1: DS 2h
304 HIBUF: DS 2h
305 BUFDST: DS 1h
306 BUFSRC: DS 1h ; IM CF-Betrieb = Anzahl der Sektoren die bertragen werden
307 CPYEND: DS 1h
308 ;
309 TIMEOUT EQU 32h ;
310 FQUARZ EQU 06h ;
311 DLY100 EQU 13h ;
312 SECZAL EQU 16 ;
313 SECLDD EQU 1 ; 256 Bytes
314 ;INTON EQU 1Eh ; RST 5.5 freigeben
315 ;INTOF EQU 1Fh ; RST 5.5 sperren
316 ;
317 ORG 2F8Ch ; Sprungtabelle fr IRQ- Serviceroutinen
318
319 intrap: ds 3h ; Software IRQ
320 int3: ds 3h ;
321 int4: ds 3h ;
322 int5: ds 3h ;
323 int55: ds 3h ; Hardware IRQ
324 int6: ds 3h ;
325 int65: ds 3h ; Hardware IRQ
326 int7: ds 3h ;
327 int75: ds 3h ; Hardware IRQ
328
329 ;
330 ;-----
331 ;
332 ; Testroutinen Datei mit Hilfe von ISR einlesen
333 ;
334 ;
335 ORG 8000h ; Testadresse
336
337 ust equ 9000h ; Userstack
338 buffer equ 9000h ; Ziel Buffer fuer BIN-Datei
339 zbuff equ 8400h ; Zeilenbuffer
340 maxbuf equ 3fh ; Bufferlaenge
341 rtsbuf equ 28h ; Bufferposition fuer RTS
342 baud equ 0040h ; fuer 4800Bd
343 ;
344 timeout equ 0600h ; ca. 2s warten
345 ;
346 ; Sprungtabelle
347 ;
348 kstart: jmp kalt ; INI
349 jmp rxzirq ; Zeilen mit IRQ lesen
350 jmp 0000h ;
351 ;
352 ;
353 ORG 8020h

```

```

354 ;
355 ; Zwischenspeicher fuer Buffer(FIFO) Steuerung
356 ; (der besseren Uebersicht hier hin verlegt)
357 ;
358 buffz dw 0000h ; Schreibzeiger fuer Zielbuffer
359 wrz dw 0000h ; Schreibzeiger fuer IRQ
360 rdz dw 0000h ; Lesezeiger
361 v24flag db 00h ; 1000 0001
362 ; :
363 ; : +-- RTS Punkt erreicht
364 ; +----- Bufferueberlauf --> Fehler
365 ;
366 ;
367 org 8040h ;
368 ;
369 kalt: lxi sp,ust ; User Stack setzen
370 call mioini ; IO-Bausteine initialisieren
371 lxi h,meld1 ;
372 call string ;
373 jmp ende ;
374 ;
375 ende: lxi h,meld10 ;
376 call string ;
377 jmp movid ; zurueck zum Monitor
378 ;
379 ;-----
380 ; Karte initialisieren
381 ;
382 mioini: mvi a,bap1 ; PIO-Betriebsarten einstellen
383 out pio1s ;
384 mvi a,bap2 ;
385 out pio2s ;
386 xra a ;
387 out pio2b ;
388 out pio2c ;
389 ;
390 mvi a,03h ; Reset com1 1
391 out com1s ;
392 mvi a,bacom1 ; com1-Betriebsart
393 out com1s ;
394 mvi a,55h ; RST On
395 out com1s ;
396 ;
397 mvi a,03h ; Reset com2 2
398 out com2s ;
399 mvi a,bacom2 ; com2-Betriebsart
400 out com2s ;
401 mvi a,55h ; RST On
402 out com2s ;
403 ;
404 mvi a,bati0 ; Timer-Betriebsarten
405 out tims ;
406 mvi a,bati1 ;
407 out tims ;
408 mvi a,bati2 ;
409 out tims ;
410 ;
411 lxi d,baud ; siehe oben
412 mov a,e ; in Timer 0 laden
413 out tim0 ;
414 mov a,d ;
415 out tim0 ;
416 ;
417 mov a,e ; in Timer 1 laden
418 out tim1 ;
419 mov a,d ;
420 out tim1 ;
421 ;
422 mov a,e ; in Timer 2 laden
423 out tim2 ;
424 mov a,d ;

```

```

425         out      tim2          ;
426         ;
427         ret
428         ;
429         ;
430         ;=====
431         ;
432         ; V24 mit IRQ Routine - Test fuer Intel-Hexlader
433         ; Daten zeilenweise empfangen und im Buffer ablegen
434         ;
435         ; Parameter zur Übersicht
436         ;
437         ; buffer      equ 9000h      ; fuer BIN-Datei
438         ; zbuff      equ 8400h      ; Buffer (FIFO)
439         ;
440         ;           v-Schreibzeiger (wrz)
441         ;           I
442         ; 8400h I-----+-----I 843fh (maxbuf)
443         ;           ^           :
444         ;           I- Lesezeiger (rdz)      842ah +----- RTS Setzen (rtsbuf)
445         ;
446         ; maxbuf      equ 3fh        ; Bufferlaenge
447         ; rtsbuf      equ 2ah        ; Bufferposition fuer RTS
448         ;
449         ; buffz       dw 0000h      ; Schreibzeiger fuer Zielbuffer
450         ; wrz         dw 0000h      ; Schreibzeiger fuer IRQ
451         ; rdz         dw 0000h      ; Lesezeiger
452         ; fuelst      db 00h        ; Anzahl byte im Buffer = Fuellstand
453         ; v24flag     db 00h        ; 1000 0001
454         ;             ; :          :
455         ;             ; :          +-- RTS punkt erreicht
456         ;             ; +----- Buffer voll
457         ;
458         ;-----
459         ;
460 rxzirq: lxi      sp,ust          ; User Stack setzen
461         lxi      h,meld11       ; Startmeldung
462         call     string         ;
463         lxi      h,buffer       ; Schreib/Lesezeiger
464         shld     buffz          ;
465         lxi      h,zbuff        ;
466         shld     wrz            ;
467         shld     rdz            ; auf Anfang
468         xra      a              ;
469         sta      v24flag        ; Testzaehler loeschen
470         call     mioini         ; Karte initialisieren
471         call     irqi           ; IRQ Serviceroutinen einrichten
472         mvi      a,95h          ; 1001 0101 = RX IRQ und
473         out      com1s          ; RTS freigeben
474         ei              ;
475         ;
476         ; Haupschleife
477         ;
478         ; Haupschleife, wartet auf Daten bis Timeout
479         ;
480 rxz_l:  push     b              ;
481 rxz_l1: lxi      b,timeout      ; Abbruchbedingung laden
482 rxz_l2: di              ;
483         lhld     wrz            ;
484         xchg     ;
485         lhld     rdz            ;
486         mov      a,l            ;
487         cmp      e              ;
488         jc       rxzrb         ; weiter, Daten lesen
489         lda      v24flag        ;
490         ana      a              ;
491         jnz      rtscl         ; RTS und IRQ freigeben
492         ei              ;
493         cpi      081h          ;
494         jz       rxz_e         ; Ende mit Bufferueberlauf
495         call     dely1          ;

```

```

496         dcr         c                ; Timeout ?
497         jnz         rxz_l2            ;
498         dcr         b                ;
499         jnz         rxz_l2            ; warten
500
501 rxz_e:    pop         b                ;
502         mvi         a,1ah             ;
503         di          ;
504         jmp         ende              ; Tiomeout keine Daten (A) = 1Ah
505 ;
506 ; Buffer auslesen
507 ;
508 rxzrb:    di          ;
509         mov         a,m                ; Byte holen
510         inx         h                ;
511         shld        rdz               ;
512         lhld        buffz            ;
513         mov         m,a                ; Byte in Zielbuffer
514         inx         h                ;
515         shld        buffz            ;
516         ei          ;
517         jmp         rxz_l1            ; warten auf naechstes Byte
518
519 ;
520 ; RTS und IRQ freigeben
521 ;
522 rtscl:    di          ;
523         lxi         h,zbuff           ; Buffer
524         shld        wrz               ;
525         shld        rdz               ; auf Anfang
526         xra         a                ;
527         sta         v24flag           ; Testzaehler loeschen
528         mvi         a,95h             ; 1001 0101 = RX IRQ und
529         out         com1s             ; RTS freigeben
530         ei          ;
531         jmp         rxz_l1            ;
532 ;
533 ;
534 ; -----
535 ;
536 ; IRQ vorbereiten
537 ;
538 irqi:     lxi         h,int65          ; Sprungtabelle
539         mvi         m,0c3h            ; = JMP
540         inx         h                ; IRQ Seviceroutine
541         lxi         d,irqs            ; In Sprungtabelle
542         mov         m,e                ;
543         inx         h                ;
544         mov         m,d                ; eintragen
545         mvi         a,1dh             ; irq6.5 freigeben
546         sim          ;
547         ret          ;
548 ;
549 ; IRQ Serviceroutine
550 ; Ein Byte uber V24 empfangen und im Empfangsbuffer ablegen
551 ;
552 irq_s:     di          ; IRQ sperren
553         push        psw               ; alles sichern
554         push        h                ;
555         push        d                ;
556         push        b                ;
557         in         com1s             ;
558         ani         01h              ; Senderegister voll
559         jz          irq_e             ; nein
560         lhld        wrz               ; Zeiger auf Zeilenbuffer
561         in         com1d             ;
562         mov         m,a                ; daten ablegen
563         inx         h                ;
564         shld        wrz               ; Zeiger sichern
565         mvi         a,rtsbuf          ; Zeiger an RTS Grenze
566         cmp         l                ;

```



```

567         jz      setrts          ; RTS Setzen
568         mvi     a,maxbuf        ; Test auf Ende Zeilenbuffer
569         cmp     l                ;
570         jz      irqbs           ; Zeilenbuffer Ende erreicht
571
572  irqe:    pop     b              ; alles wieder zurueck
573         pop     d                ;
574         pop     h                ;
575         pop     psw             ;
576         ei      ; IRQ freigeben
577         ret                    ;
578
579  setrts:  mvi     a,0d5h         ; RTS gesetzt
580         out     comls           ; IRQ Freigabe
581         mvi     a,01h          ;
582         sta     v24flag         ;
583         jmp     irqe           ;
584
585  irqbs:   lxi     h,zbuff        ; Buffer wieder auf Anfang
586         shld    wrz            ;
587         mvi     a,55h          ; RTS gesetzt
588         out     comls           ; IRQ gesperrt
589         mvi     a,81h          ;
590         sta     v24flag         ;
591         jmp     irqe           ;
592 ;
593 ;-----
594 ;
595 ; Texte
596 ;
597  meld1:   DB 0Dh                ; CR
598         DB 0Ah                ; LF
599         DB "Multi-IO-Test"
600         DB " V0.11 INI"
601         DB 0Dh                ;
602         DB 00h
603 ;
604
605  meld10:  DB 0Dh                ; CR
606         DB 0Ah                ; LF
607         DB "Ende"
608         DB 0Dh                ;
609         DB 00h
610 ;
611  meld11:  DB 0Dh                ; CR
612         DB 0Ah                ; LF
613         DB "Multi-IO-Test"
614         DB " V0.11 "
615         DB "Datei lesen"
616         DB 0Dh                ;
617         DB 00h
618 ;
619 ;-----
620
621

```