

```

;
2: ; Filnavn: NEWBURN.ASM
3: ; Besk.:PIC16x84 seriel programmer
4: ; modtager med 57600 baud, 8 bit, 1 stopbit
5: ; ingen paritet og ingen flowcontrol
6: ; Passer til 10Mhz krystal
7: ; Dato: 29. Marts 2006
8: ;
9:
10: processor 16f84a ;Dette direktiv angiver hvilken processor MPASM skal genererer kode til
11: #include p16f84a.inc ;Dette direktiv angiver at en filen "p16f84a.inc" skal indsættes på dette sted
12:
13:
14: ; Sætter configuration-bits ved at anvende definitioner fra include-filen, p16f84a.inc
15: __config _HS_OSC & _PWRTE_OFF & _WDT_OFF & _CP_OFF
16:
17: #define clock portb,6 ;definitioner
18: #define dataP portb,7 ;definitioner
19: #define mclr porta,3 ;definitioner
20: #define reset porta,2 ;definitioner
21: #define clockT Trisb,6 ;definitioner
22: #define dataT trisb,7 ;definitioner
23: #define mclrT trisa,3 ;definitioner
24: #define resetT trisa,2 ;definitioner
25: #define stat porta,1 ;definitioner
26: #define statT trisa,1 ;definitioner
27: #define over2000 flagreg,0 ;hvis PC er over 2000hex
28: #define over2100 flagreg,1 ;hvis PC er over 2100hex
29:
30: counter equ 0x0C ;Delay til RS232
31: BitCounter equ 0x0D ;Delay til RS232
32: Receive equ 0x0E ;Det modtagne tegn fra RS232
33: transmit equ 0x0F ;Tegn til afsendelse på RS232
34: counter1 equ 0x10 ;Delay_10ms routine
35: counter2 equ 0x11 ;Delay_10ms routine
36: flagreg equ 0x12 ;register til flag
37: prog_count equ 0x13 ;counter til antal byte på hexlinien
38: prog_adr0 equ 0x14 ;programadresse (lav)
39: prog_adr1 equ 0x15 ;programadresse (høj)
40: prog_type equ 0x16 ;type indikerer sidste linie i hexfilen
41: prog_con0 equ 0x17 ;configurations-word (lav)
42: prog_con1 equ 0x18 ;configurations-word (høj)
43: prog_PicA0 equ 0x19 ;adressen som der brændes på i PIC kredsen (lav)
44: prog_picA1 equ 0x1A ;adressen som der brændes på i PIC kredsen (høj)
45: prog_data0 equ 0x1B ;databyte (lav) fra hexlinien
46: prog_data1 equ 0x1C ;databyte (høj) fra hexlinien
47: prog_tdata0 equ 0x1D ;databyte (lav) til at sammenligne med
48: prog_tdata1 equ 0x1F ;databyte (høj) til at sammenligne med
49: prog_bit_count equ 0x20 ;bit-counter til seriel-routinerne til pic'en
50: prog_cmd_data equ 0x21 ;plads til at gemme cmd i
51: col equ 0x22 ;tæller til x-retning
52: lin equ 0x23 ;tæller til y-retning
53: chksum equ 0x24 ;checksum
54: extra equ 0x25 ;en extra plads
55: flashEE equ 0x26 ;fflash = 0 og EE = 1
56: PrgMem equ 0x27 ;størrelsen af Program memory
57: DatMem equ 0x28 ;størrelsen af Data memory (EE)
58:
59: org 0x0000 ;Her starter prog. afviklingen (reset vektor)
60: clrf counter ;delay til RS232
61: clrf counter1 ;Delay 10ms
62: clrf counter2 ;Delay 10ms
63: goto main ;gå til hovedprogrammet
64: org 0x0005 ;første adr. efter interrupt vektor
65:
66: #include RS232.lib ;indsæt RS232 bibliotek
67: main ;**** hovedprogram ****
68: bcf mclr ;mclr lav
69: bcf clock ;clock lav
70: bcf dataP ;data lav
71: bcf stat ;fejllindikator lav
72: bcf reset ;normal reset høj

```

```

73:
74:      bsf      status,rp0      ;definer porte
75:      bsf      clockT          ;indgang
76:      bsf      dataT           ;indgang
77:      bcf      mclrT          ;udgang
78:      bcf      statT          ;udgang
79:      bcf      resetT         ;udgang
80:      bcf      status,rp0     ;definition slut
81:
82:      clrf     prog_PicA0      ;\ Reset Pic-
83:      clrf     prog_PicA1      ;/ adresse tæller
84:      bcf      over2000        ;PC er ikke over 2000hex
85:      bcf      over2100        ;PC er ikke over 2100hex
86:
87:      call     Receive_port    ;init RS232 modtag
88:      call     transmit_port   ;init RS232 send
89:
90:      call     delay_10ms      ;\
91:      call     delay_10ms      ; pause
92:      call     delay_10ms      ;/
93:
94:      movlw   'R'              ; Skriv et 'READY' på rs232
95:      movwf   transmit         ; "
96:      call    Transmit_data    ; "
97:      movlw   'e'              ; "
98:      movwf   transmit         ; "
99:      call    Transmit_data    ; "
100:     movlw   'a'              ; "
101:     movwf   transmit         ; "
102:     call    Transmit_data    ; "
103:     movlw   'd'              ; "
104:     movwf   transmit         ; "
105:     call    Transmit_data    ; "
106:     movlw   'y'              ; "
107:     movwf   transmit         ; "
108:     call    Transmit_data    ; "
109:     movlw   D'13'            ; (CR)"
110:     movwf   transmit         ; "
111:     call    Transmit_data    ; "
112:     movlw   D'10'            ; (LF) "
113:     movwf   transmit         ; "
114:     call    Transmit_data    ; "
115: endloop                       ;**** vent på kolon ****
116:     clrf     receive         ;slet indholdet i receive-reg.
117:     clrwf   receive         ;slet indholdet i w-reg.
118:     call    receive_wait    ;vent på et tegn på rs232 (tegnet lander i "receive")
119:     movf    receive,w       ;Flyt tegnet til W (igen)
120:     SUBLW  ':'               ;sammenlign med kolon
121:     BTFSC  status,Z        ;var det et kolon?
122:     GOTO   start_burn      ;hvis ja: start brændingen
123:
124:     movf    receive,w       ;Flyt tegnet til W (igen)
125:     SUBLW  '#'              ;sammenlign med #
126:     BTFSC  status,Z        ;var det et #?
127:     GOTO   read_pic        ;hvis ja: læs PIC'en
128:
129:     movf    receive,w       ;Flyt tegnet til W (igen)
130:     SUBLW  '§'              ;sammenlign med §
131:     BTFSC  status,Z        ;var det et §?
132:     GOTO   slet_pic        ;hvis ja: slet PIC'en
133:
134:     movf    receive,w       ;Flyt tegnet til W (igen)
135:     SUBLW  '¤'              ;sammenlign med ¤
136:     BTFSC  status,Z        ;var det et ¤?
137:     GOTO   picburnll       ;hvis ja: send ID og et "Ready"
138:
139:     goto    endloop        ;hvis nej: bliv i loopet
140: ;*****
141: picburnll
142:     call    read_ID         ;læs hvilken PIC som er tilsluttet brænderen
143:     call    send_ID        ;send det til pc-programmet
144:     GOTO   main            ;afslut og send et "Ready"

```

```

145: read_ID
146:      call    enterPrgMem      ;gå ind i brænde-mode
147:      call    delay_10ms      ;pause
148:      movlw   0xFF            ;flyt et tilfældigt tal til W-reg
149:      movwf   prog_data0      ;put det i adr0
150:      movwf   prog_data1      ;put det i adr1
151:      movlw   D'0'            ;load config cmd
152:      call    prog_cmd         ;send cmd til pic'en
153:      call    prog_send_data   ;send data til pic'en
154:      call    flyt_til_2006    ;flyt til adr. 2006
155:      movlw   D'4'            ;læs data fra prog-mem cmd
156:      call    prog_cmd         ;send cmd til pic'en
157:      call    read_adr         ;læs adresseindhold ind i prog_data0 og prog_data1
158:      MOVLW   0xE             ;
159:      SUBWF   prog_data1,w     ;
160:      BTFSS   status,Z        ;
161:      BSF     flashEE,1       ;flash = 0 og EE = 1
162:      BTFSC   status,Z        ;
163:      BCF     flashEE,1       ;
164:      return
165: send_ID
166:      movlw   prog_data1      ;\ send heigh-byte's
167:      call    write_ASCII     ;/ på RS232
168:      movlw   prog_data0      ;\ send low-byte's
169:      call    write_ASCII     ;/ på RS232
170:      return
171: ;*****
172: read_pic      ;**** Læs indholdet ***
173:      movlw   PrgMem          ;\ læs hvor meget prog. mem der (kommer fra pc-programmet)
174:      call    read_hex        ;/
175:      movlw   DatMem          ;\ læs hvor meget data. mem der (kommer fra pc-programmet)
176:      call    read_hex        ;/
177:
178:      call    enterPrgMem     ;gå ind i brænde-mode
179:      call    delay_10ms     ;pause
180:      clrf   prog_adr0       ;adr-tæller=0 (low)
181:      clrf   prog_adr1       ;adr-tæller=0 (heigh)
182:
183: PRloop      movlw   0x80     ;læg 128 i w-reg ### skal rettes hvis data er større end 1024 byte ###
184:      movwf   lin            ;flyt værdi til y-tæller
185: loop_lin
186:      movlw   D'8'           ;læg 8 i w-reg
187:      movwf   col            ;flyt værdi til x-tæller
188:      call    prefix         ;send prefix
189:      call    chkstatus      ;opdater checksum
190: loop_col
191:      movlw   D'4'           ;læs data fra prog-mem cmd
192:      call    prog_cmd       ;send cmd til pic'en
193:      call    read_adr       ;læs adresseindhold ind i prog_data0 og prog_data1
194:      movlw   prog_data0     ;\ send low-byte's
195:      call    write_ASCII    ;/ på RS232
196:      movlw   prog_data1     ;\ send heigh-byte's
197:      call    write_ASCII    ;/ på RS232
198:      movf    prog_data1,w   ;w-reg=data1
199:      addwf   chksum,f       ;læg til check sum
200:      movf    prog_data0,w   ;w-reg=data0
201:      addwf   chksum,f       ;læg til check sum
202:      incf   prog_adr0,f     ;forøg adr tæller (lav)
203:      btfsc  status,Z       ;(p.g.a. 16 bit)
204:      incf   prog_adr1,f     ;forøg evt. også høj del
205:      incf   prog_adr0,f     ;forøg adr tæller (lav)
206:      btfsc  status,Z       ;(p.g.a. 16 bit)
207:      incf   prog_adr1,f     ;forøg evt. også høj del
208:      movlw   D'6'           ;forøg adr i pic komandoen
209:      call    prog_cmd       ;send cmd til pic'en
210:      decfsz col             ;tæl ned på x-tæller
211:      goto   loop_col        ;hvis ikke 0 fortsæt i loop
212:      call    l_f            ;send checksum og linefeed
213:      decfsz lin            ;forminsk y-tæller
214:      goto   loop_lin        ;hvis ikke 0 fortsæt i loop
215:      decfsz PrgMem         ;forminsk PrgMem-tæller
216:      goto   PRloop         ;hvis ikke 0 fortsæt i loop

```

```

217:                                     ;**** flyt til adresse 2000hex ****
218:      movlw      0xFF                                     ;flyt et tilfældigt tal til W-reg
219:      movwf     prog_data0                             ;put det i adr0
220:      movwf     prog_data1                             ;put det i adr1
221:      movlw     D'0'                                    ;load config cmd
222:      call      prog_cmd                               ;send cmd til pic'en
223:      call      prog_send_data                         ;send data til pic'en
224:      movlw     0x40                                    ;\
225:      movwf     prog_adr1                              ; sæt adresse til 4000h
226:      clrf      prog_adr0                              ;/
227:      movlw     ':'                                     ;en linie starter med kolon
228:      movwf     transmit                               ;flyt w-reg til transmit
229:      call      transmit_data                          ;send på RS232
230:      movlw     '0'                                    ;så kommer et 0
231:      movwf     transmit                               ;flyt w-reg til transmit
232:      call      transmit_data                          ;send på RS232
233:      movlw     '8'                                    ;og dernæst et 8
234:      movwf     transmit                               ;flyt w-reg til transmit
235:      call      transmit_data                          ;send på RS232
236:      call      prefix1                               ;hent resten af prefix
237:      movlw     D'8'                                    ;w-reg=8
238:      movwf     chksum                                ;flyt til check sum
239:      movf      prog_adr1,w                            ;w-reg= adr1
240:      addwf     chksum,f                               ;læg til check sum
241:      movf      prog_adr0,w                            ;w-reg= adr0
242:      addwf     chksum,f                               ;læg til check sum
243:      movlw     D'4'                                    ;læg 4 i w-reg
244:      movwf     col                                    ;flyt værdi til x-tæller
245:      call      loop_id                                ;send data
246:      call      l_f                                    ;send chksum og linefeed
247:      movlw     D'6'                                    ;\
248:      call      prog_cmd                               ; \
249:      movlw     D'6'                                    ;  \ flyt til adr
250:      call      prog_cmd                               ;  / 2007hex
251:      movlw     D'6'                                    ; /
252:      call      prog_cmd                               ;/
253:      movlw     0x0E                                    ;\ sæt adresse
254:      movwf     prog_adr0                              ;/ til 2007hex
255:      movlw     ':'                                     ;en linie starter med kolon
256:      movwf     transmit                               ;flyt w-reg til transmit
257:      call      transmit_data                          ;send på RS232
258:      movlw     '0'                                    ;så kommer et 0
259:      movwf     transmit                               ;flyt w-reg til transmit
260:      call      transmit_data                          ;send på RS232
261:      movlw     '2'                                    ;og dernæst et 2
262:      movwf     transmit                               ;flyt w-reg til transmit
263:      call      transmit_data                          ;send på RS232
264:      call      prefix1                               ;hent resten af prefix
265:      movlw     D'2'                                    ;w-reg=2
266:      movwf     chksum                                ;flyt til check sum
267:      movf      prog_adr1,w                            ;w-reg= adr1
268:      addwf     chksum,f                               ;læg til check sum
269:      movf      prog_adr0,w                            ;w-reg= adr0
270:      addwf     chksum,f                               ;læg til check sum
271:      movlw     D'1'                                    ;læg 1 i w-reg
272:      movwf     col                                    ;flyt værdi til x-tæller
273:      call      loop_id                                ;send data
274:      call      l_f                                    ;send chksum og linefeed
275:      bcf      status,C                               ;\
276:      rrf      prog_adr1,f                             ; divider prog_adrX med 2
277:      rrf      prog_adr0,f                             ;/ (da 4000hex i filen = 2000hex i pic'en)
278:      incf     prog_adr0,f                             ;forøg med 1
279: mov2100
280:      movlw     D'6'                                    ;forøg -
281:      call      prog_cmd                               ;pic-adressen med 1
282:      incf     prog_adr0,f                             ;forøg adr tæller (lav)
283:      btfsz    status,Z                               ;(p.g.a. 16 bit)
284:      incf     prog_adr1,f                             ;forøg evt. også høj del
285:      movlw     0x21                                    ;læg 21hex i w-reg
286:      subwf    prog_adr1,w                             ;sammenlign med adr1
287:      btfsz    status,Z                               ;defferens?
288:      goto     mov2100                                ;hvis forskel: forøg

```

```

289:          bcf      status,C          ;\
290:          rlf      prog_adr0,f       ; adr x 2
291:          rlf      prog_adr1,f       ; /
292:
293: DAloop    movlw    D'8'              ;sæt gentagelse til 8 ## skal rettes hvis EE-data er større end 64 byte ##
294:          movwf    lin                ;flyt værdien til lin
295: EEdataX
296:          call     prefix              ;send liniestart
297:          call     chkstatus           ;opret checksum
298:          call     EEmem              ;send data
299:          call     l_f                ;send chksum og linefeed
300:          movlw    0x10              ;w-reg=10h
301:          addwf    prog_adr0,f        ;læg w-reg til adr (low)
302:          decfsz   lin                ;forminsk y-tæller
303:          goto     EEdataX           ;hvis ikke nul bliv da i loopet
304:          decfsz   DatMem             ;forminsk DatMem-tæller
305:          goto     DAloop            ;hvis ikke nul bliv da i loopet
306:          movlw    ':'                ;**** send sidste linie :00000001FF ****
307:          call     transmit_data      ;
308:          movlw    '0'               ;
309:          call     transmit_data      ;
310:          movlw    '0'               ;
311:          call     transmit_data      ;
312:          movlw    '0'               ;
313:          call     transmit_data      ;
314:          movlw    '0'               ;
315:          call     transmit_data      ;
316:          movlw    '0'               ;
317:          call     transmit_data      ;
318:          movlw    '0'               ;
319:          call     transmit_data      ;
320:          movlw    '0'               ;
321:          call     transmit_data      ;
322:          movlw    '1'               ;
323:          call     transmit_data      ;
324:          movlw    'F'               ;
325:          call     transmit_data      ;
326:          movlw    'F'               ;
327:          call     transmit_data      ;
328:          movlw    D'13'             ;
329:          call     transmit_data      ;
330:          movlw    D'10'             ;
331:          call     transmit_data      ;
332:          goto     main              ;læsning færdig
333: ;-----
334: loop_id   movlw    D'4'              ;læs data fra prog-mem cmd
335:          call     prog_cmd           ;send cmd til pic'en
336:          call     read_adr           ;læs adresseindhold ind i prog_data0 og prog_data1
337:          movlw    prog_data0        ;\ send low-byte's
338:          call     write_ASCII        ;/ på RS232
339:          movlw    prog_data1        ;\ send heigh-byte's
340:          call     write_ASCII        ;/ på RS232
341:          movf     prog_data1,w       ;w-reg=data1
342:          addwf    chksum,f           ;læg til check sum
343:          movf     prog_data0,w       ;w-reg=data0
344:          addwf    chksum,f           ;læg til check sum
345:          movlw    D'6'              ;forøg adr i pic komandoen
346:          call     prog_cmd           ;send cmd til pic'en
347:          decfsz   col                ;tæl ned på x-tæller
348:          goto     loop_id           ;hvis ikke 0 fortsæt i loop
349:          return                    ;vend tilbage
350: ;-----
351: EEmem     movlw    D'8'              ;gentagelsesfaktor
352:          movwf    col                ;lægges i col
353: loop_EE   movlw    D'5'              ;læs data fra prog-mem cmd
354:          call     prog_cmd           ;send cmd til pic'en
355:          call     read_adr           ;læs adresseindhold ind i prog_data0 og prog_data1
356:          clrf     prog_data1        ;sæt data (heigh) til 00h
357:          movlw    prog_data0        ;\ send low-byte's
358:          call     write_ASCII        ;/ på RS232
359:          movlw    prog_data1        ;\ send heigh-byte's
360:          call     write_ASCII        ;/ på RS232

```

```

361:          movf    prog_data1,w    ;w-reg=data1
362:          addwf   chksum,f        ;læg til check sum
363:          movf    prog_data0,w    ;w-reg=data0
364:          addwf   chksum,f        ;læg til check sum
365:          movlw   D'6'            ;forøg adr i pic komandoen
366:          call    prog_cmd        ;send cmd til pic'en
367:          decfsz  col             ;tæl ned på x-tæller
368:          goto    loop_EE        ;hvis ikke 0 fortsæt i loop
369:          return                   ;vend tilbage
370: ;-----
371: l_f          comf    chksum,f      ;2's komplement af checksum
372:          incf    chksum,f        ;forøg med 1
373:          movlw   chksum         ;send checksum
374:          call    write_ASCII    ;på RS232
375:          movlw   D'13'         ;\
376:          movwf   transmit      ; send CR
377:          call    Transmit_data  ;/
378:          movlw   D'10'         ;\
379:          movwf   transmit      ; send LF
380:          call    Transmit_data  ;/
381:          return
382: ;-----
383: prefix      ;**** linie-prefix ****
384:          movlw   ':'            ;en linie starter med kolon
385:          movwf   transmit      ;flyt w-reg til transmit
386:          call    transmit_data  ;send på RS232
387:          movlw   '1'          ;så kommer et 1
388:          movwf   transmit      ;flyt w-reg til transmit
389:          call    transmit_data  ;send på RS232
390:          movlw   '0'          ;og dernæst et 0
391:          movwf   transmit      ;flyt w-reg til transmit
392:          call    transmit_data  ;send på RS232
393: prefix1     movlw   prog_adr1   ;hent adr (heigh)
394:          call    write_ASCII    ;send på RS232
395:          movlw   prog_adr0     ;hent adr (low)
396:          call    write_ASCII    ;send på RS232
397:          movlw   '0'          ;type 00
398:          movwf   transmit      ;flyt w-reg til transmit
399:          call    transmit_data  ;send på RS232
400:          movlw   '0'          ;type 00
401:          movwf   transmit      ;flyt w-reg til transmit
402:          call    transmit_data  ;send på RS232
403:          return                   ;retuner
404: ;-----
405: chkstatus   ;**** opdater chksum ***
406:          movlw   D'16'         ;2's komplement af antal og type
407:          movwf   chksum        ;flyt til check sum
408:          movf    prog_adr1,w    ;2's komplement af adr1
409:          addwf   chksum,f      ;læg til check sum
410:          movf    prog_adr0,w    ;2's komplement af adr0
411:          addwf   chksum,f      ;læg til check sum
412:          return                   ;returner
413: ;-----
414: write_ASCII ;**** send register som 2 ASCII ****
415:          movwf   fsr           ;peg på filereg (pointer)
416:          swapf  indf,f        ;byt om på 4 øverste og 4 nederste bit
417:          call    sendNib       ;send en ASCII-karakter på RS232
418:          swapf  indf,f        ;byt tilbage igen (og send en ASCII til)
419: sendNib     movf    indf,w      ;flyt destination til W-reg
420:          andlw  B'00001111'    ;læg 48 til (svarer til ASCII-nul)
421:          addlw  D'48'          ;flyt resultat til transmit-reg
422:          movwf  extra         ;flyt resultat til transmit-reg
423:          movlw  D'58'          ;
424:          subwf  extra,w        ;
425:          btfsz  status,C       ;
426:          goto   sendchr       ;
427:          movlw  D'7'          ;
428:          addwf  extra,f        ;
429:          ;
430: sendchr     movf    extra,w     ;
431:          movwf  transmit      ;flyt w-reg til transmit
432:          ;

```



```

433:          call      transmit_data      ;send ASCII
434:          return     ;vend tilbage
435: ;-----
436: read_adr          ;*** læs en adresse ind i 2 reg's ***
437:          movlw     D'16'                ;put 16 i w-reg.
438:          movwf    prog_bit_count       ;sæt bit-count til 16
439:          bsf      status,rp0           ;\
440:          bsf      dataT                 ; sæt dataport til input
441:          bcf      status,rp0           ;/
442:          nop                      ;\
443:          nop                      ; \ pause
444:          nop                      ;/
445:          nop                      ;/
446: read_adr1         ;**** loop ****
447:          bsf      clock                ;clock=1
448:          nop                      ;pause
449:          nop                      ;pause
450:          bcf      clock                ;data klar til læsning
451:          nop                      ;pause
452:          btfscc   dataP                 ;\
453:          bsf      status,C              ; \ sæt C = dataport
454:          btfscc   dataP                 ;/
455:          bcf      status,C              ;/
456:          rrf      prog_data1,f          ;roter C ind i data1
457:          rrf      prog_data0,f          ;roter data1 ind i data0
458:          nop                      ;pause for at være sikker
459:          decf     prog_bit_count        ;forminsk bit-counter
460:          btfscc   status,Z              ;er den nul?
461:          goto     read_adr1             ;hvis nej: fortsæt loopet
462:          bsf      status,rp0           ;\
463:          bcf      dataT                 ; hvis ja: sæt dataport til output
464:          bcf      status,rp0           ;/
465:          bcf      prog_data0,0          ;\reset start-
466:          bcf      prog_data1,7          ;/og stop-bit
467:          bcf      status,C              ;sæt carry = 0
468:          rrf      prog_data1,f          ;roter C ind i data1
469:          rrf      prog_data0,f          ;roter data1 ind i data0
470:          return     ;afslut og vend tilbage (se *)
471:          ;*(prog_data0=adr(low) og prog_data1=adr(heigh)
472: ;*****
473: slet_pic         ;**** Slet Pic'en ****
474:          call     enterPrgMem          ;gå ind i brænde-mode
475:          call     delay_10ms           ;pause
476:          call     sletPic              ;slet PIC-kredsen
477:          goto     main                 ;sletning færdig
478: ;*****
479: start_burn      ;**** start brænding ****
480:          call     read_ID              ;##### sat ind i forb. med udvikling
481:          bcf      stat                 ;fejllindikator lav
482:          call     enterPrgMem          ;gå ind i brænde-mode
483:          call     delay_10ms           ;pause
484:          call     sletPic              ;anbefalet sletning af pic
485: LinieStart     ;**** læs en linie i hexfilen ****
486:          movlw    prog_count           ;\ hent count
487:          call     read_hex              ;/ fra hexlinien
488:          movlw    prog_adr1            ;\ hent high-byte adr
489:          call     read_hex              ;/ fra hexlinien
490:          movlw    prog_adr0            ;\ hent low-byte adr
491:          call     read_hex              ;/ fra hexlinien
492:          bcf      status,C              ;\
493:          rrf      prog_adr1,f           ; divider prog_adrX med 2
494:          rrf      prog_adr0,f           ;/ (da 4000hex i filen = 2000hex i pic'en)
495:          movlw    prog_type            ;\ hent type
496:          call     read_hex              ;/ fra hexlinien
497:          movlw    D'1'                 ;er det type 01 ?, hvilket betyder
498:          subwf    prog_type,w           ;sidste linie...
499:          btfscc   status,Z              ;hvis ja - så...
500:          goto     burnconf             ;brænd config og afslut brænding
501:          movlw    0x07                 ; hvis dette er config-word,
502:          subwf    prog_adr0,w           ; så gem det til sidst og
503:          btfscc   status,Z              ; fortsæt...
504:          goto     ikke2007             ;

```

```

505:          movlw    0x20          ;
506:          subwf   prog_adr1,w   ;
507:          btfss   status,Z      ;
508:          goto    ikke2007      ;
509:          call    hent2byte     ;
510:          movf    prog_data0,w  ;
511:          movwf   prog_con0     ;
512:          movf    prog_data1,w  ;
513:          movwf   prog_con1     ;
514:          incf    prog_picA1,f  ;
515:          decf    prog_count,f  ;
516:          btfsc   status,Z      ;
517:          goto    prog_skip     ;
518:          decf    prog_count,f  ;
519:          btfsc   status,Z      ;
520:          goto    prog_skip     ;
521:          btfss   over2000     ;
522:          goto    loadconf     ;
523:          call    opdatPicAdr   ;
524:          goto    prog_word     ;
525: ikke2007          ;fortsæt her hvis adressen ikke er 2007
526:          btfsc   over2000     ;hvis adresse er over 2000
527:          goto    check2100    ;så flyt PC til configurations-
528:          movlw   0x20         ;memory....
529:          subwf   prog_adr1,w  ;
530:          btfsc   status,Z      ;
531:          goto    loadconf     ;
532: check2100        ;fortsæt her hvis adressen ikke er 2000
533:          btfsc   over2100     ;hvis adresse er over 2100
534:          goto    continue     ;så gå i EEdata-mode....
535:          movlw   21h         ;
536:          subwf   prog_adr1,w  ;
537:          btfsc   status,Z      ;
538:          call    reenterPrgMem ;
539: continue          ;fortsæt her hvis adressen ikke er 2100
540:          call    opdatPicAdr   ;synkroniser pic-adr med prog-adr
541: Prog_word        ;**** programmer 2 byte ****
542:          call    hent2byte     ;hent 2 byte fra hexlinien
543:          movlw   D'2'         ;cmd load data for program mem
544:          btfsc   over2100     ;hvis EEdata-mode
545:          addlw   D'1'         ;så: cmd load data for data mem
546:          call    prog_cmd     ;send cmd til pic'en
547:          call    prog_send_data ;send data tilpic'en
548: hex2000          ;start her hvis: kom fra loadcon
549:          call    burn_word     ;brænd ordet
550:          decf    prog_count,f  ;forminsk count
551:          btfsc   status,z      ;hvis nul så gå til prog_skip
552:          goto    prog_skip     ;(evt. fejlcheck!!)
553:          decf    prog_count,f  ;forminsk count
554:          btfss   status,Z      ;hvis nul så gå til prog_skip
555:          goto    prog_word     ;ellers fortsæt med at læse byte fra hexlinien
556: prog_skip        ;**** læs bytes frem til kolon ****
557:          call    receive_wait  ;hent en byte fra hexlinien
558:          movlw   ':'          ;læg et kolon i w
559:          subwf   receive,w     ;træk det fra den læste byte
560:          btfss   status,Z      ;var byten et kolon?
561:          GOTO    prog_skip     ;hvis nej, hent da en ny byte fra hexlinien
562:          goto    liniestart    ;hvis ja, gå da igang med en ny hexlinie
563: loadconf        ;**** første gang man når til config-mem ****
564:          call    hent2byte     ;hent 2 byte fra hexlinien
565:          movlw   D'0'         ;load config cmd
566:          call    prog_cmd     ;send cmd til pic'en
567:          call    prog_send_data ;send data til pic'en
568:          movlw   0x00         ;\
569:          movwf   prog_picA0   ; \ sæt pic adr
570:          movlw   0x20         ; / til 2000hex
571:          movwf   prog_picA1   ;/
572:          call    opdatPicAdr   ;synkroniser pic-adr med prog-adr
573:          bsf    over2000     ;vi er nu i config-Mem området
574:          goto    hex2000     ;spring til prog_word rutinen
575: opdatPicAdr      ;**** synkroniser adresser ****
576:          goto    cmp_picAdr   ;start med at sammenlign prog-adr og pic-adr

```



```

577: inc_picAdr          ;***** forøg *****
578:      movlw          D'6'          ;forøg adr i pic komandoen
579:      call           prog_cmd       ;send cmd til pic'en
580:      incf           prog_PicA0,f    ;forøg pic adr tæller (lav)
581:      btfsz          status,Z        ;(p.g.a. 16 bit)
582:      incf           prog_PicA1,f    ;forøg evt. også høj del
583: cmp_picAdr          ;***** sammenlign *****
584:      movf           prog_adr1,w     ;\
585:      subwf          prog_PicA1,w    ; sammenlign høj del
586:      btfsz          status,Z        ;/
587:      goto          inc_picAdr       ; hvis forskel: forøg
588:      movf           prog_adr0,w     ;\
589:      subwf          prog_PicA0,w    ; sammenlign lav del
590:      btfsz          status,Z        ;/
591:      goto          inc_picAdr       ; hvis forskel: forøg
592:      return
593: hent2byte           ;**** læs 2 bytes fra hexfilen ****
594:      movlw          prog_data0      ;i dette register skal byten ligge
595:      call           read_hex        ;hent byten fra hexlinien
596:      movlw          prog_data1      ;i dette register skal byten ligge
597:      call           read_hex        ;hent byten fra hexlinien
598:      return
599: burn_word           ;**** brænd og sammenlign ****
600:      BTFSS         flashEE,1
601:      goto          bnfl
602:
603:      movlw          B'001000'       ;Denne sekvens hvis PIC'en er en EEPROM*****
604:      call           prog_cmd       ;cmd begin program
605:      call           delay_10ms      ;send cmd til pic'en
606:      goto          bw_nxt          ;vent 10 ms
607: bnfl                ;gå til sammenlign
608:      movlw          B'011000'       ;Denne sekvens hvis PIC'en er en FLASH*****
609:      call           prog_cmd       ;cmd begin program only
610:      call           delay_1ms       ;send cmd til pic'en
611:      movlw          B'10111'       ;vent 1ms
612:      call           prog_cmd       ; \ end programming
613: bw_nxt              ; /
614:      movlw          D'4'            ;read data from prog-mem cmd
615:      btfsz          over2100        ;hvis EEdata-mode så:
616:      addlw          D'1'            ;read data from data mem cmd
617:      call           prog_cmd       ;send cmd til pic'en
618:      call           prog_compare    ;sammenlign det sendte og det læste
619:      movlw          D'6'            ;increment adrs cmd
620:      call           prog_cmd       ;send cmd til pic'en
621:      incf           prog_picA0,f    ;forøg pic adr tæller (lav)
622:      btfsz          status,Z        ;(p.g.a. 16 bit)
623:      incf           prog_picA1,f    ;forøg evt. også høj del
624:      return
625: prog_cmd            ;**** send prog-komando ****
626:      movwf          prog_cmd_data   ;gem kommando der skal sendes
627:      movlw          D'6'            ;put 6 i w-reg.
628:      movwf          prog_bit_count  ;gem 6 i bit-counter
629: prog_cmd1          ;**** loop ****
630:      bsf           clock           ;clock=1
631:      rrf           prog_cmd_data,f  ;roter cmd_data og få bit i c
632:      btfsz          status,C        ;\
633:      bsf           dataP           ; \ sæt data udgangen
634:      btfsz          status,C        ; / til det samme som C
635:      bcf           dataP           ;/
636:      nop           ;\
637:      nop           ; pause for at være sikker
638:      nop           ;/
639:      bcf           clock           ;clock=0
640:      nop           ;\ pause for at
641:      nop           ;/ være sikker
642:      decf          prog_bit_count   ;forminsk bit-counter
643:      btfsz          status,Z        ;er den nul?
644:      goto          prog_cmd1       ;hvis nej: fortsæt loopet
645:      return                    ;hvis Ja: afslut og vend tilbage
646: prog_send_data     ;**** Send 16 bit data ****
647:      bcf           status,C        ;sæt C = 0
648:      rlf           prog_data0,f    ;\

```

```

649:                rlf      prog_data1,f      ; læg 0 i start og slut
650:                bcf      prog_data1,7      ;/
651:                movf     prog_data0,w      ;\
652:                movwf    prog_tdata0      ; \ gem de 2 data bytes
653:                movf     prog_data1,w      ; / i Tdata 1 og 2
654:                movwf    prog_tdata1      ;/
655:                movlw    D'16'            ;put 16 i w-reg.
656:                movwf    prog_bit_count    ;gem 16 i bit-counter
657: prog_send1      ;**** loop ****
658:                bsf      clock            ;clock=1
659:                nop      ;pause
660:                nop      ;pause
661:                rrf      prog_data1,f      ;roter data1 over i data0
662:                rrf      prog_data0,f      ;roter data0 og få bit i C
663:                btfs    status,C          ;\
664:                bsf      dataP            ; \ sæt data udgangen
665:                btfs    status,C          ; / til det samme som C
666:                bcf      dataP            ;/
667:                nop      ;pause for at være sikker
668:                bcf      clock            ;clock=0
669:                nop      ;pause for at være sikker
670:                nop      ;pause for at være sikker
671:                decf     prog_bit_count    ;forminsk bit-counter
672:                btfs    status,Z          ;er den nul?
673:                goto    prog_send1        ;hvis nej: fortsæt loopet
674:                return   ;hvis Ja: afslut og vend tilbage
675: prog_compare     ;*** sammenlign sendte data med modtaget data ***
676:                movlw    D'16'            ;put 16 i w-reg.
677:                movwf    prog_bit_count    ;sæt bit-count til 16
678:                bsf      status,rp0       ;\
679:                bsf      dataT            ; sæt dataport til input
680:                bcf      status,rp0       ;/
681:                nop      ;\
682:                nop      ; \ pause
683:                nop      ;/
684:                nop      ;/
685: prog_cmp1        ;**** loop ****
686:                bsf      clock            ;clock=1
687:                nop      ;pause
688:                nop      ;pause
689:                bcf      clock            ;data klar til læsning
690:                nop      ;pause
691:                btfs    dataP            ;\
692:                bsf      STATUS,C         ; \ sæt C = dataport
693:                btfs    dataP            ; /
694:                bcf      STATUS,C         ;/
695:                rrf      prog_data1,f      ;roter C ind i data1
696:                rrf      prog_data0,f      ;roter data1 ind i data0
697:                nop      ;pause for at være sikker
698:                decf     prog_bit_count    ;forminsk bit-counter
699:                btfs    STATUS,Z          ;er den nul?
700:                goto    prog_cmp1        ;hvis nej: fortsæt loopet
701:                bsf      STATUS,rp0       ;\
702:                bcf      dataT            ; hvis ja: sæt dataport til output
703:                bcf      STATUS,rp0       ;/
704:                bcf      prog_data0,0     ;\reset start og stop bit
705:                bcf      prog_data1,7     ;/før sammenligning
706:                movf     prog_data0,w      ;\sammenlign
707:                subwf    prog_Tdata0,w    ;/low-byte
708:                btfs    status,Z          ;var de ens?
709:                goto    Cmp_error        ;hvis nej: fejl!!
710:                btfs    over2100         ;hvis ja: er det EEdata?
711:                call    comp_high        ;hvis nej: sammenlign high-byte
712:                return   ;hvis ja: afslut og vend tilbage
713: comp_high        ;**** sammenlign high-byte ****
714:                movf     prog_data1,w      ;\sammenlign
715:                subwf    prog_Tdata1,w    ;/high-byte
716:                btfs    status,Z          ;var de ens?
717:                goto    Cmp_error        ;hvis nej: fejl!!
718:                return   ;hvis ja: afslut og vend tilbage
719: Cmp_error        ;**** fejl routine ****
720:                movlw    prog_picA1      ; \ send heigh-byte's

```

```

721:      call      write_ASCII      ;/ på RS232
722:      movlw    prog_picA0        ;\ send low-byte's
723:      call      write_ASCII      ;/ på RS232
724:      movlw    'E'                ;Skriv et 'Error' på rs232
725:      movwf    transmit          ;
726:      call      Transmit_data     ;
727:      movlw    'r'                ;
728:      movwf    transmit          ;
729:      call      Transmit_data     ;
730:      movlw    'r'                ;
731:      movwf    transmit          ;
732:      call      Transmit_data     ;
733:      movlw    'o'                ;
734:      movwf    transmit          ;
735:      call      Transmit_data     ;
736:      movlw    'r'                ;
737:      movwf    transmit          ;
738:      call      Transmit_data     ;
739:      bsf      stat               ;indiker fejl på LED
740:      return                      ;afslut og vend tilbage
741: reenterPrgMem                    ;**** vend tilbage til prg-mem ****
742:      bsf      over2100           ;vis EEdata-mode
743:      clrf    prog_picA0         ;reset pic-adressen
744:      clrf    prog_picA1         ;
745:      nop                          ;pause
746:      nop                          ;pause
747: enterPrgMem                      ;**** gå ind i prg-mem ****
748:      bcf      mclr               ;\reset pic'en
749:      bsf      reset             ;/
750:      call    delay_10ms         ;pause
751:      bcf      clock             ;sæt clock til 0
752:      bcf      dataP             ;sæt data til 0
753:      bsf      status,rp0        ;sæt clock og data til output
754:      bcf      clockT            ;
755:      bcf      dataT             ;
756:      bcf      status,rp0        ;
757:      bcf      clock             ;sæt clock til 0
758:      bcf      dataP             ;sæt data til 0
759:      nop                          ;pause
760:      nop                          ;
761:      nop                          ;
762:      bcf      reset             ;gå i brænde mode
763:      bsf      mclr               ;
764:      nop                          ;pause
765:      nop                          ;
766:      nop                          ;
767:      return                      ;afslut og vend tilbage
768: burnconf                          ;*** brænd configuratiop og afslut brænding ***
769:      call    reenterPrgMem       ;gå ind i prog-mode
770:      bcf      over2100           ;men ikke i EEdata-mode
771:      movf    prog_con0,w         ;
772:      movwf   prog_data0         ;hent configword (lav)
773:      movf    prog_con1,w         ;
774:      movwf   prog_data1         ;hent configword (høj)
775:      movlw   D'0'                ;load config cmd
776:      call    prog_cmd            ;send cmd til pic'en
777:      call    prog_send_data      ;send data til pic'en
778:      call    flyt_til_2007       ;flyt til adr 2007
779:      movlw   D'8'                ;#####
780:      call    prog_cmd            ;#####
781:      call    delay_10ms         ;#####
782:      goto    main                ;brændingen færdig
783: flyt_til_2007                      ;forøger fra adr 2000hex
784:      movlw   D'6'                ;til 2007hex...
785:      call    prog_cmd            ;
786: flyt_til_2006                      ;(start her hvis det er ID, der skal læses)
787:      movlw   D'6'                ;
788:      call    prog_cmd            ;
789:      movlw   D'6'                ;
790:      call    prog_cmd            ;
791:      movlw   D'6'                ;
792:      call    prog_cmd            ; (7 X incr-adr cmd)

```

```

793:      movlw    D'6'      ;
794:      call     prog_cmd   ;
795:      movlw    D'6'      ;
796:      call     prog_cmd   ;
797:      movlw    D'6'      ;
798:      call     prog_cmd   ;
799:      return    ;
800: ;***** slet pic'en *****
801: sletPic
802:      clr     prog_data0   ;procedure til at slette pic'en
803:      comf    prog_data0,f ;
804:      clr     prog_data1   ;
805:      comf    prog_data1,f ;
806:      movlw    D'0'      ;
807:      call     prog_cmd   ;
808:      call     prog_send_data ;
809:      call     flyt_til_2007 ;
810:      movlw    D'1'      ;
811:      call     prog_cmd   ;
812:      movlw    D'7'      ;
813:      call     prog_cmd   ;
814:      movlw    D'8'      ;
815:      call     prog_cmd   ;
816:      call     delay_10ms  ;
817:      movlw    D'1'      ;
818:      call     prog_cmd   ;
819:      movlw    D'7'      ;
820:      call     prog_cmd   ;
821:      movlw    B'11111'   ;
822:      call     prog_cmd   ;
823:      call     delay_10ms  ;
824:      call     reenterPrgMem ;
825:      bcf     over2100    ;
826:      return    ;
827: ;***** RechHex*****
828: Read_Hex
829:      movwf    fsr        ;sæt pointer til at modtage hexbyte
830:      clr     indf        ;slet destination
831:      call     Nibble     ;Modtag 4 bit i et hextegn
832:      swapf   indf,f     ;swap det til highnibble
833:      goto    nibble     ;Modtag 4 bit igen
834: Nibble
835:      call     receive_wait ;vent på et tegn på seriel port
836:      movlw    D'48'      ;hex - 48 (værdien af '0')
837:      subwf   receive,f   ;
838:      bcf     receive,5   ;evt. korriger for smaa bogstaver i hex
839:      movlw    D'7'      ;evt -7
840:      btsc    receive,4   ;test for større end 16 ('A'..'F')
841:      subwf   receive,f   ;Her er recieve 0..15
842:      mov     receive,w   ;
843:      iorwf   indf,f     ;OR resultatet ind i INDR destination
844:      return    ;
845: ;*****burner delay routine (EEPROM-kredse)*****
846: Delay_10ms ; 10,19ms = (772*21h+6)*400ns
847:      clr     counter1    ;nulstil tæller 1
848:      movlw    0x21      ;21h i w-reg.
849:      movwf   counter2    ;læg 21h i tæller 2
850: loop10ms ;**** loop ****
851:      decfsz  counter1,F  ;forminsk tæller 1 (resultat i tæller)
852:      goto    loop10ms    ;hvis tæller ikke er nul: fortsæt loop
853:      decfsz  Counter2,F  ;ellers: forminsk tæller 2 (resultat i tæller)
854:      goto    loop10ms    ;hvis tæller ikke er nul: fortsæt loop
855:      return    ;ellers: afslut og vend tilbage
856: ;*****burner delay routine (FLASH-kredse)*****
857: Delay_1ms ; 1,23ms = (772*4h+6)*400ns
858:      clr     counter1    ;nulstil tæller 1
859:      movlw    0x4       ;4h i w-reg.
860:      movwf   counter2    ;læg 4h i tæller 2
861: loop1ms ;**** loop ****
862:      decfsz  counter1,F  ;forminsk tæller 1 (resultat i tæller)
863:      goto    loop1ms     ;hvis tæller ikke er nul: fortsæt loop
864:      decfsz  Counter2,F  ;ellers: forminsk tæller 2 (resultat i tæller)

```

```
865:          goto      loop1ms          ;hvis tæller ikke er nul: fortsæt loop
866:          return     ;ellers: afslut og vend tilbage
867: ;-----
868:          end          ;programmet er slut!!!
```