

```

1  *BEGIN
2  !
3  *INTEGERFN VOLT(*INTEGER CHAN,GAIN)
4  LACC(CHAN<<8+GAIN<<3)
5  *EMT_363
6  *RESULT=ACC
7  *END !! FN VOLT
8  !
9  *REALFN RTIME
10 *INTEGER A,B
11 *MOV_R1,-(SP)
12 *EMT_321 !! TIME IN R0,R1
13 *MOV_R1,R5
14 *MOV_(SP)+,R1;!!RESTORE R1
15 *MOV_R0,34.(R1);!!HIGH ORDER IN A
16 *MOV_R5,36.(R1);!!LOW ORDER IN B
17 A=A+1 *IF B<0 !!SIGN BIT?
18 *RESULT=(A*65536.+B)/3000 !! MINUTES
19 *END !!OF RTIME
20 !
21 *ROUTINE SETPOT(*INTEGER X,UNIT)
22 !
23 ! X=UNIT => ON, X=0 => OFF
24 !
25 ! RELAY9 -- START COUNTING
26 ! RELAY10 -- RESET TIMER
27 !
28 *MOV_R1,-(SP)
29 *MOV_14.(R1),R0 !! X
30 *MOV_16.(R1),R1 !! UNIT
31 *EMT_352
32 *MOV_(SP)+,R1
33 *END !! SETPOT
34 !
35 *INTEGERFN IABS(*INTEGER I)
36 I=-I *IF I<0
37 *RESULT=I
38 *END !!IABS
39 !
40 *REALFN ABS(*REAL B)
41 B=-B *IF B<0
42 *RESULT=B
43 *END !ABS
44 !
45 *ROUTINE WAIT(*INTEGER X)
46 !WAITS X/50 SECS AND THEN RETURNS
47 !X SHOULD NOT EXCEED 32,767
48 LACC(X)
49 *EMT_375
50 *END
51 !
52 *ROUTINE RINGBELL(*INTEGER I)
53 *INTEGER K
54 *CYCLE K=1,1,I
55 PRINT SYMBOL(7)
56 ! BELL IS CTRL G ON TT
57 *REPEAT
58 *END!! OF RINGBELL
59 !
60 *REALFN MAKEREAL(*INTEGER AD)
61 *REAL X
62 *MOV_14.(R1),R2
63 *MOV_#36.,R3
64 *ADD_R1,R3

```

```

65      *MOV_(R2)+,(R3)+
66      *MOV_(R2)+,(R3)+
67      *RESULT=X
68      *END;!! MAKEREAL
69      !
70      *ROUTINESPEC PRINTFL(XREAL X,XINTEGER N)
71      *REALFN MOD(XREAL X)
72      * _BIC_#100000,14,(R1)
73      *RESULT=X
74      *END;!! OF FN MOD
75      *ROUTINE PRINT(XREAL X,XINTEGER N,M)
76      *DOWNINTEGER IMAX=32767; !MAX INTEGER IN ONE WORD.
77      *REAL Y,Z,ROUND
78      *INTEGER I,J,L,SIGN,DP;      ! DP=DECIMAL POINT
79      *4 *IF M>4; DP=0;      ! DEAL WITH STUPID PARAMS
80      *IF N<0 *THEN N=1;      ! DEAL WITH STUPID PARAMS
81      *M=MOD(X);      ! ALL WORK DONE WITH Y
82      *ROUND= (1/2)/10**M;      ! ROUNDING FACTOR
83      *IF Y>IMAX *THEN *START
84      *PRINTFL(X,M); *RETURN; *FINISH
85      *I=0;Z=1; Y=Y+ROUND;
86      *COUNT LEADING PLACES:
87      *I=I+1;Z=10*Z;      ! NO DANGER OF OVERFLOW HERE
88      *IF Y>Z *THEN -> COUNT LEADING PLACES
89      *SPACES(N-I);      ! O.K FOR ZERO OR -VE SPACES
90      *SIGN=' ';      ! ** IMPLIED
91      *IF X<0 *THEN SIGN='- '
92      *PRINT SYMBOL(SIGN)
93      *J=I-1; Z=10**J
94      *NEXT DIGIT:
95      *L=INT PT(Y/Z);      ! OBTAIN NEXT DIGIT
96      *Y=Y-L*Z;Z=Z/10;      ! AND REDUCE TOTAL
97      *PRINT SYMBOL(L+'0')
98      *J=J-1
99      *IF J>=0 *THEN -> NEXT DIGIT
100     *IF M=0 *THEN *RETURN;      ! NO DECIMAL PART TO BE O/P
101     *IF DP=0 *THEN *PRINTTEXT'.'
102     *DP=1; J=0; Z=1
103     *Y=10*Y; M=M-1; -> NEXT DIGIT
104     *END
105     *ROUTINE PRINTFL(XREAL X,XINTEGER N)
106     *REAL SIGN,ROUND,FACTOR
107     *INTEGER COUNT,INC
108     *N=4 *IF N>4
109     *ROUND=.5/10**N
110     *SIGN=1
111     *IF X=0 *THEN -> ZERO
112     *IF X < 0 *THEN*START
113     *X=-X; SIGN=-SIGN; *FINISH
114     *INC=1; COUNT=0; FACTOR=1/10
115     *IF X <= 1 *THEN*START
116     *FACTOR=10; INC=-1; *FINISH
117     *SCALE: *IF 1<=X+ROUND *AND X+ROUND<10 *THEN -> PRINTOUT
118     *X=X*FACTOR; COUNT=COUNT+INC
119     *-> SCALE
120     *ZERO: COUNT=-99
121     *PRINTOUT: PRINT(SIGN*X,1,N)
122     *PRINTTEXT'0'
123     *WRITE(COUNT,2)
124     *END
125     *ROUTINE READF(*REALNAME XX)
126     *INTEGER ADR
127     * _MOV_14,(R1),36,(R1); ! ADR=ADDR(XX)
128     !

```

```

129      !
130      *ROUTINESPEC READ(XINTEGER ADR,PARM)
131      *ROUTINESPEC SKIP SYMBOL
132      *INTEGERFENSPEC NEXT SYMBOL
133      *SWITCH SW(1:6)
134      *INTEGER CH,P,PT,DIG,STATE
135      *INTEGERARRAY L(1:72)
136      *OWNINTEGERARRAY A(0:48)= XC
137      2, -1, -1, 5, -1, -1, -1,
138          2, 2, 3, 5, 6, -1, -1,
139          1, 0, 0, 4, 0, 0, 0,
140          1, 0, 0, -1, 0, 0, 0,
141          -1, -1, 4, -1, -1, -1, -1,
142          3, 3, -1, -1, -1, -1, -1,
143          -1, -1, -1, -1, -1, -1, -1
144      *INTEGER ROW
145      -> FIRST
146      RESET:
147      *PRINTTEXT 'WRONG';NEWLINE
148      FIRST:
149      P=0
150      STATE=1
151      INSCH:
152      READ SYMBOL(CH)
153      P=P+1
154      L(P)=CH
155      ROW=0
156      *IF CH='+' *OR CH='-' *THEN -> OUT
157      ROW=ROW+1
158      *IF '0'<=CH *AND CH<='9' *THEN -> OUT
159      ROW=ROW+1
160      *IF CH='.' *THEN -> OUT
161      ROW=ROW+1
162      *IF CH='10' *THEN -> OUT
163      ROW=ROW+1
164      *IF CH='2' *THEN -> OUT
165      ROW=ROW+1
166      *IF CH='.' *THEN -> OUT
167      ROW=ROW+1
168      OUT:
169      STATE=A(7*ROW+STATE-1)
170      -> INSCH *IF STATE>0
171      -> RESET *IF STATE<0
172      PT=1
173      READ(ADR,1)
174      !
175      !
176      *ROUTINE READ(XINTEGER ADR,PARM)
177      *INTEGER FLAG,CURSYM; ! FLAG= 0FOR'-' ,1 FOR '+'
178      *INTEGER IVALUE,J
179      *REAL RWORK,SCALE,WK2
180      *OWNREAL IMAX=32767.0
181      FLAG=1
182      -> TEST SIGN
183      IGNORE LEADING SPACES:
184      SKIP SYMBOL
185      TEST SIGN:CURSYM=NEXT SYMBOL; ! CARE NOT TO READ TERMINATOR
186      -> IGNORE LEADING SPACES *IF CURSYM='.' *OR CURSYM=10
187      -> PASS SIGN *IF CURSYM='+'
188      -> DIGIT *UNLESS CURSYM='-'
189      MINUS: FLAG=0; ! RECORD INITIAL MINUS
190      PASS SIGN: SKIP SYMBOL; ! MOVE OVER SIGN ONCE IT HAS
191      CURSYM=NEXT SYMBOL; ! BEEN RECORDED IN FLAG

```

```

192 DIGIT: -> DIGIT NOT FIRST XUNLESS '0'<=CURSYM XAND CURSYM<='9'
193 RWORK=CURSYM-'0'; ! KEEP TOTAL IN RWORK
194 SKIP SYMBOL
195 LOOP: CURSYM=NEXT SYMBOL
196 -> NOT DIG XUNLESS '0'<=CURSYM XAND CURSYM<='9'
197 RWORK=10*RWORK+(CURSYM-'0'); ! CONTINUE EVALUATING
198 -> LOOP
199 NOT DIG: -> TRY AT XUNLESS CURSYM='.'
200 SCALE=10
201 FPART: SKIP SYMBOL
202 CURSYM=NEXT SYMBOL
203 -> TRY AT XUNLESS '0'<=CURSYM XAND CURSYM<='9'
204 RWORK=RWORK+(CURSYM-'0')/SCALE
205 SCALE=10*SCALE; -> FPART
206
207 TRY AT:
208 ! THE VALUE HAS NOW BEEN READ INTO RWORK. THERE MIGHT BE AN EXPONENT
209 ! E.G. '1.78 10' IS VALID DATA FOR READ
210 -> FIX XUNLESS CURSYM='e'
211 SKIP SYMBOL; ! MOVE PAST THE 'e'
212 READ( ADDR(IVALUE),0); ! RECURSIVE CALL TO FIND EXPONENT
213 XIF IVALUE<=-99 XTHENSTART
214 RWORK=0
215 -> FIX
216 XFINISH
217 WK2=10.0
218 XIF IVALUE<0 XTHENSTART
219 IVALUE=-IVALUE
220 WK2=0.1
221 XFINISH
222 -> FIX XIF IVALUE=0
223 XCYLE J=1,1,IVALUE
224 RWORK=RWORK*WK2
225 XREPEAT
226
227 FIX:
228 ! KNOCK NUMBER INTO RIGHT FORM
229 -> INT READ XIF PARM=0
230 XIF FLAG=0 XTHEN RWORK=-RWORK
231 !REAL(ADR)=RWORK
232 !J=ADDR(RWORK)
233 !INTEGER(ADR)=INTEGER(J)
234 !INTEGER(ADR+2)=INTEGER(J+2)
235
236 *_MOV_#46.,R0
237 *_ADD_R1,R0 ; ! R0=ADDR(RWORK)
238 *_MOV_14.(R1),R3
239 *_MOV_(R0)+,(R3)+
240 *_MOV_(R0)+,(R3)+
241 XRETURN
242
243 INT READ:
244 ! IVALUE= INT(RWORK)
245 WK2=0.5
246 XIF RWORK<0 XTHEN WK2=-0.5
247 IVALUE=INTPT(RWORK*WK2)
248 XIF FLAG=0 XTHEN IVALUE=-IVALUE
249 !INTEGER(ADR)=IVALUE
250 *_MOV_42.(R1),@14.(R1)
251
252 XRETURN
253 DIGIT NOT FIRST:
254 ! CAN HAVE .73 AS VALID IMP NO
255 SKIP SYMBOL
256 CURSYM=NEXT SYMBOL
257 RWORK=(CURSYM-'0')/10
258 SCALE=100; -> FPART
259 XEND; ! READ
260
261 XROUTINE SKIPSYMBOL

```

```

256          PT=PT+1
257          XEND; ! SKIP SYMBOL
258          %INTEGERFN NEXT SYMBOL
259          %RESULT=L(PT)
260          XEND; ! NEXT SYMBOL
261          XEND; ! READF
262          %ROUTINE READ(%INTEGERNAME I)
263          %REAL X
264          ! USE READF FOR INTEGERS SINCE IT IS IMMUNE TO SYMBOL IN DATA
265          READF(X)
266          %IF X>30000. %THENSTART ; ! INTPT FAILS FOR X>2**15
267          I=30000;%RETURN;%FINISH
268          X=X-1 %IF X<0 ;!COMPENSATE FOR ERROR IN INTPT
269          I=INTPT(X+.001)
270          XEND
271          !
272          %REALFN SORT(%REAL X)
273          %REAL Y,Z
274          %IF X<=0 %THENSTART
275          %IF Y=0 %THEN %RESULT=0
276          NEWLINE;%PRINTTEXT* NEG ARG IN SORT*
277          ;NEWLINE
278          %RESULT=1.
279          %FINISH
280          Y=(1+X)/2
281          I:Z=(Y+X/Y)/2
282          %IF Z>=Y %THEN %RESULT=Y
283          Y=Z
284          ->I
285          XEND;!! OF FN SORT
286          !
287          ! DIRECT ACCESS FILE ROUTINES
288          !
289          %ROUTINE CREATE(%INTEGERARRAYNAME FILE)
290          LACC(ADDR(FILE(2)))
291          *MOV_R1,-(SP)
292          *MOV_(R0),R2
293          *MOV_-(R0),R1
294          *MOV_-(R0),R0
295          *EMT_345 ;! CREATE
296          *MOV_(SP)+,R1 ;! RESTORE R1
297          XEND ;! CREATE
298          %INTEGERFN FIRSTBLOCK(%INTEGERARRAYNAME FILE)
299          LACC(ADDR(FILE(2)))
300          *MOV_R1,-(SP)
301          *MOV_(R0),R2
302          *MOV_-(R0),R1
303          *MOV_-(R0),R0
304          *EMT_344 ;! UDEX
305          *MOV_(SP)+,R1 ;! RESTORE R1
306          %RESULT=ACC
307          XEND ;! FIRSTBLOCK
308          %INTEGERFN NEXTBLOCK(%INTEGER BLOCK)
309          LACC(BLOCK)
310          *MOV_R1,R5
311          *EMT_340
312          *MOV_R5,R1
313          %RESULT=ACC
314          XEND ;! NEXTBLOCK
315          %INTEGERFN FILESIZE(%INTEGERARRAYNAME FILE)
316          %INTEGER I,J
317          I=0
318          J=FIRSTBLOCK(FILE)
319          I: %RESULT=I %IF J=0 XOR I=4000

```

```

320      I=I+1
321      J=NEXTHBLOCK(J)
322      ->1
323      XEND !! FILESIZE
324      !
325      %INTEGERFN BLOCKNOF(%INTEGER FIRST,N)
326      %INTEGER I
327      %RESULT=0 %IF N<1 %OR N>4000
328      I=FIRST
329      1:%IF N=1 %OR I=0 %THEN %RESULT=I
330      I=NEXTHBLOCK(I)
331      N=N-1
332      ->1
333      XEND !! BLOCKNOF
334      !
335      !
336      %OWNINTEGER DABLOCK
337      %OWNINTEGER DAPROTEC
338      %ROUTINE OPENDA(%INTEGERARRAYNAME FILE)
339      ! OPEN FOR READ OR WRITE
340      1:LACC(ADDR(FILE(2)))
341      *MOV_R1,R5
342      *MOV_(R0),R2
343      *MOV_-(R0),R1
344      *MOV_-(R0),R0
345      *EMT_344 !! UDEX
346      *MOV_R1,-(SP)
347      *MOV_R5,R1 !! RESTORE R1
348      DABLOCK=ACC
349      *MOV_(SP)+,R0
350      DAPROTEC=ACC
351      %RETURNIF DABLOCK#0
352      CREATE(FILE)
353      ->1
354      XEND!! OPENDA
355      %ROUTINE WRITEDA(%INTEGER BLOCK,X)
356      ! X IS ADDRESS OF START OF 512 BYTE BLOCK
357      %INTEGER DWBLOCK,I,J
358      %IF BLOCK<0 %OR BLOCK>200 %THENSTART
359      %PRINTTEXT* BAD BLOCK NUMBER*;%NEWLINE
360      *EMT_372
361      %STOP;%FINISH
362      %IF DABLOCK=0 %THENSTART
363      %PRINTTEXT* DAFILE NOT OPEN*;%NEWLINE
364      *EMT_372
365      %STOP;%FINISH
366      2:DWBLOCK=BLOCKNOF(DABLOCK,BLOCK)
367      %IF DWBLOCK=0 %THENSTART!!FILE TOO SMALL
368      LACC(DAPROTEC)
369      *MOV_P0,-(SP)
370      LACC(DABLOCK)
371      *MOV_R1,R5
372      *MOV_(SP)+,R1
373      *EMT_341 !! APPEND
374      *MOV_R5,R1
375      ->2
376      %FINISH
377      LACC(X)
378      !ADDRESS
379      *MOV_R0,-(SP)
380      LACC(DAPROTEC)!! PROTEC
381      *MOV_R0,-(SP)
382      LACC(DWBLOCK)!! BLOCK NUMBER
383      *MOV_R1,R5

```

```

384      *MOV_(SP)+,R3
385      *MOV_#-400,R2
386      *MOV_(SP)+,R1
387      *EMT_330!! WRITE
388      *MOV_R5,R1
389      XEND !! WRITEDA
390      !
391      !
392      ! TEXT FILE OUTPUT
393      !
394      XROUTINE DEFINE OUTPUT(XINTEGERARRAYNAME A)
395          LACC(ADDR(A(0)))
396          *MOV_#UTFIL,R5
397          *MOV_(R0)+,(R5)+
398          *MOV_(R0)+,(R5)+
399          *MOV_(R0)+,(R5)+
400          XEND !!DEFINEOUTPUT
401      !
402      !
403      XROUTINESPEC COLLECTDATA
404      XROUTINESPEC SETOUTPUT
405      XROUTINESPEC GETDATA
406      XROUTINESPEC DECODE
407      XROUTINESPEC WRITECODE
408      XROUTINESPEC STOPPROG
409      XROUTINESPEC WRITETODISC
410      XROUTINESPEC ANALYSE(XREALARRAYNAME D1,XREAL M1,LIM,XREALNAME MEAN,XC
SIGMA,XINTEGERNAME N1)
411      XROUTINESPEC WAITPROG
412      XROUTINESPEC CHECKINTERRUPT
413      XROUTINESPEC LOADVARS
414      XROUTINESPEC INITIALISE
415      XROUTINESPEC WRITESPEC
416      XROUTINESPEC MONITOR
417      XROUTINESPEC DRIVEGRATING
418      XROUTINESPEC SUMMARISE(XINTEGER STEP)
419      XROUTINESPEC CHECKERROR
420      XREALFNSPEC READLASER
421      XINTEGER I,P,CTIME,COUNTS,COUNTS0,COUNTS1,RINT,MINT,MONIT,T1,DTIME
422      XINTEGER ERRC0,NSETS,ERRCD1,IMONIT,SFLAG,IFLAG
423      XINTEGER FLAG1,FLAG2,FLAG3,FLAG4
424      XINTEGER ISET,STEPS,SEQNO,IINST,ISTEP,NC,NC1,NC2,NC0,QFLAG,IE,RTI
425      XREAL MM,LL,TIMEIN,TIMEOUT
426      XINTEGER NBLOCK,FBLOCKS,SPECNO
427      XINTEGER OLDDIR,NEWDIR
428      XREALARRAY P1(1:16)
429      XREALARRAY PP1(1:16)
430      XREALARRAY P0(1:16)
431      XREALARRAY PP0(1:16)
432      XREALARRAY ILASER(1:50)
433      XREAL LPOW,LPOW1,LSIGMA,LFAC1,LFAC2,LSCALE
434      XREAL MPO,MP1,MPP0,MPP1,Q0,Q1,Q00,Q01
435      XREAL MTIME,DNOISE,SUMERR,FFAULT,FTIME
436      XINTEGERARRAY A(0:11)
437      XINTEGERARRAY STORE(0:11)
438      XINTEGERARRAY S0(1:16)
439      XINTEGERARRAY S1(1:16)
440      XINTEGERARRAY FNAME(0:2)
441      XINTEGERARRAY TNAME(0:2)
442      XINTEGERARRAY NACC(-50:50)
443      XREALARRAY SIGNAL(-50:50)
444      XREALARRAY ERROR(-50:50)
445      XREALARRAY EXPVAR(0:31)
446      XOWNINTEGERARRAY REPLY(1:9)='K','W','C','R','D','F','B','U','S'

```