rtl

RTL/2

FORTRAN Interface Capability For P800 DOS

RTL/2 FORTRAN Interface Capability For Philips P800 under DOS

RTL/2 Reference : 78 Version : 2

Philips Data Systems Publ. Nr. 5122 991 28121

Author : G.C. Stevenson

Date : 10th March 1977

C O N T E N T S

C O N T E N T S

SECTION 1 INTRODUCTION

1.1 Purpose of this document

This document describes a special additional phase which can be added to RTL/2 compilers to allow FORTRAN subprograms to be called from RTL/2 modules.

1.2 The "Middle End"

The production of special calling sequences for RTL/2 - to - FORTRAN interfacing is performed by an additional compiler phase. This phase is run after the Front End and before the Back End and hence is referred to as a "Middle End".

1.3 Generality

Although the Middle End is inevitably target machine dependent, and concerned with the coding styles of both FORTRAN and RTL/2 on the target machine, it has been structured in such a way that most of it is portable. The middle end might also be enhanced to generate interface code for RTL/2 calls to procedures written in languages other than FORTRAN.

1.4 The Structure of this Document

This document reflects the middle end structure and philosophy; that is, some sections are target machine specific, while others are general.

SECTION 2 GENERAL DESCRIPTION

2.1 Compiler Operation

An RTL/2 compiler with the middle end operates as far as possible with the same operating instructions as one without the middle end.

When the FORTRAN interface facility is not in use there is no difference in the compiler output and only a minimal compilation time overhead associated with the middle end.

2.2 Requesting FORTRAN Interfacing

(See Also Section 4)

The user asks for FORTRAN interface code generation by supplying an extra parameter

FOR = Y ⌈ES⌉

Refer to Section 4 for precise details, which are host machine and operating system dependent.

2.3 Specifying FORTRAN procedures

Procedures which are FORTRAN subroutines or functions are specified in the RTL/2 source text simply as RTL/2 EXT PROCs.

When the FOR = Y parameter is specified, the compiler will search the RTL/2 source for a list of EXTPROC names which are in fact FORTRAN written. The names are separated by commas, spaces, newlines or percent signs (%) and an ampersand (&) must appear immediately before the first name and immediately after the last name. The information must be in the form of an RTL/2

comment to avoid causing front end errors and the information must not be preceded by any other ampersand.

2.4 Making use of FORTRAN procedures

The RTL/2 text can make normal RTL/2 use of FORTRAN procedures. An RTL/2 procedure variable may be set to point to a FORTRAN procedure and the correct interfacing code will be executed when the procedure variable is called.

As usual a procedure defined as returning a result can be used for side-effects only, the result being ignored.

2.5 Call-by-reference and call-by-value

RTL/2 offers only call-by-value. Call-by-reference effect is obtained by call-by-value with REF parameters. This is true also of FORTRAN calls from RTL/2.

For example, if a particular FORTRAN subroutine, FSUBA, takes a single integer parameter then the RTL/2 user of it may define either:-

```
%1% EXT PROC (INT) FSUBA;
```

in which case a copy of the INTeger parameter is made for FSUBA and the caller's data is immune from FSUBA actions, or

```
%2% EXT PROC (REF INT) FSUBA;
```

in which case FSUBA is using a pointer to some RTL/2 variable to which it can thus assign.

2.6 Middle End Mechanism

The middle end has two main areas of work. Firstly it seeks references to any of the FORTRAN procedures to 'bend' these to be references to interface procedures. Then it adds the interface procedures to the semi-compiled program.

SECTION 3 TARGET MACHINE INFORMATION

3.1 P800 Coding Styles

Users are referred to:-

> RTL/2 Reference 69, "The RTL/2 Run Time Environment on the Philips P800 Series" (5122 991 28131)

for details of RTL/2 coding style, and to the Philips P800 FORTRAN manual for FORTRAN (5122 991 11401).

3.2 Main Incompatibilities

3.2.1 Call Instruction

FORTRAN uses CF A14,...
whereas RTL/2 uses CF A6,...

3.2.2 Register Usage

FORTRAN routines may use any registers (except A15 which is reserved for system use, and A14 which is preserved as a stack pointer.)
RTL/2 expects A12 and A13 to be preserved across procedure calls.

3.2.3 Parameters

The following parameter types are supported in the middle end:-

```
      INT
      REAL
REF   ARRAY  BYTE   )
REF   ARRAY  INT    )   (Single dimension only)
REF   ARRAY  REAL   )
      REF INT
      REF REAL
```

3.2.4 Result passing

Both FORTRAN and RTL/2 results are passed in registers using A1 onwards.

3.3 Interface Code Generated

3.3.1 Calls on FORTRAN procedures

The middle end changes references to EXT PROCs which have been specified to be FORTRAN to be references to local procedures. These references include the static and dynamic setting of procedure variables as well as literal procedure calls.

3.3.2 Local Interface Procedures

The local procedures are interface procedures; one interface procedure is generated for each FORTRAN procedure name which is not only quoted in the FORTRAN names list (and specified in the RTL/2 module as an EXT PROC) but also is referenced within the RTL/2 module.

The name of an interface procedure is the corresponding FORTRAN name with a '#' prefixed.

3.3.3 Interface Details

Each interface procedure consists of procedure entry and exit control routine calls, between which is a CODE statement.

The CODE contains instructions to:-

(1) Save registers A12, A13.

(2) Reformat the parameters into a FORTRAN argument list with A4 pointing to the list.

(3) Call the FORTRAN routine.

(4) Restore registers A12, A13.

3.3.4 Use of FORTRAN names in CODE statements

The normal philosophy of CODE statements applies i.e. it is the user's responsibility to use FORTRAN names correctly where they appear in CODE statements. Bending of the reference takes place

if the name follows a trip-1 character.
Thus if RSINE were a FORTRAN procedure a CODE statement may contain

(a) &RSINE

in which case it is translated to be a reference to the generated interface procedure brick, #RSINE,

or

(b) RSINE

in which case a direct reference to the actual FORTRAN routine is made, unbeknown to the RTL/2 compiler.

SECTION 4 RUNNING THE COMPILER

4.1 Operating Instructions

Users are referred to:-

RTL/2 Reference 73, "Philips P800
DOM RTL/2 User Manual"
(5122 991 28111)

for details of the command strings used to drive the RTL/2 compiler.

The compiler with the middle end may be used transparently when no FORTRAN interfacing is to be performed; all existing command files can continue to be used without conversion.

4.2 Using the FORTRAN interface facility

The middle end is activated by an additional parameter, FOR = Y [ES]

If this parameter is specified, the middle end will search the RTL/2 source for a & sign. It will then read the names of the FORTRAN subprograms, separated by a comma, a % sign or a new line until it encounters another & sign. Obviously, this information should be enclosed in % signs, to make it an RTL/2 comment, otherwise an RTL/2 syntax error will occur. The comment should be near the start of the program, to save time when the compiler is searching through the source and because the information must not be preceded by any other ampersand (&).

4.3 Example

Compile as

```
$RTL  FN=PROG, LS=2, XREF=X:PROG, FOR=Y
```

The source module, PROG, will have a comment near the beginning something like this:

```
%FORTRAN NAMES ARE &FSUBA,FSUBB%
%FSUBC,FSUBD&%
```

4.4 Environment Errors

Any of the environment errors described in the RTL/2 User Manual may occur.

SECTION 5 REPORT FORMAT

5.1 Fortran Interface Report

The middle end contributes the following to the Compiler listing/report output.

(a) The title:-

FORTRAN SUBPROGRAM REPORT

(b) When the source program is being listed, a line numbered listing of the FORTRAN names as input.

(c) Error messages of the form:-

FORTRAN NAME "xxxxx" ERROR nnn
For details of error numbers, nnn, refer to Section 6. "xxxxxx" is the FORTRAN subprogram name associated with the error.

(d) Messages of the form:-

FORTRAN NAME "xxxxxx" PROCESSED LINE nnn
indicating where references to FORTRAN subprograms have been processed by the middle end.

5.2 Other differences from standard compiler report

5.2.1 Brick Names and Sizes

The bricks generated by the middle end (with names beginning with '#') appear in the brick size table.

SECTION 6 ERRORS DETECTED BY THE FORTRAN INTERFACE

6.1 Front and back-end errors

All the usual errors detected by the front-end and back-end of the compiler are possible.

When an error (other than a warning) is detected by the front-end, neither the back-end nor the middle end will be run. When an error (other than a warning) is detected by the middle end, the back-end will not be run.

6.1.1 Back End Error 1 may occur in particular, if there is insufficient space for the generation of FORTRAN interface code.

6.2 Environmental Errors

See Section 4.

6.3 Middle End Errors

Two classes of error are distinguished warnings and serious errors. Warnings have error numbers greater than 200 and do not affect subsequent execution. Serious errors have numbers from 100 to 200 and will prevent running of the back end.

Table 6.3.1 Middle End Errors

Error No.	Meaning
Errors	
102	Name in FORTRAN names list is not an EXT PROC.
103	Too many FORTRAN names (Maximum = 256).
104	FORTRAN procedure with unsupported parameter or result type.
105	Compiler tables cannot accommodate additional names for interface procedures.
106	Compiler tables cannot accommodate additional name characters for interface procedures.
108	Supposedly FORTRAN brick present in RTL/2 module!
Warnings	
201	Name specified in FORTRAN names list does not appear in RTL/2 module.

SECTION 7 EXAMPLE

This is a program which actually runs and the FORTRAN and RTL/2 components are both illustrated.

```
$RTL FN=CALLSF,LS=2,FOR=Y,INFO=1,OP=KPF
MOV CALLSF,/S
REF /D4
RDA /BF,/D4
ASG /BD,DK
ASG /BE,DK
MOV RTL,/L,SAG,/FD
```

```
RUN
DATE  09 /03 /77   TIME 10H-45M-34S-

LABEL = SPLCOPY            DATE =  18.6.76          PACK NBR =    W      RTL2


RTL/2 ON PHILIPS P800 - S.P.L. VERSION 1.1

    0   IDENT CALLSF
    1  LET VT=11;
    2  LET NL=10;
    3  LET FF=12;
    4  SVC DATA RRSIO;
    5     PROC() BYTE IN;
    6     PROC(BYTE) OUT;
    7  ENDDATA;
    8
    9  % FORTRAN NAMES ARE &NPRIME,ZZZALG,SINEG&%
   10  EXT PROC(REF ARRAY BYTE )TWRT;
   11  EXT PROC (INT) PROC ()BYTE RROPI;
   12  EXT PROC (INT) PROC (BYTE) RROPO,RROPOS;
   13  EXT PROC (PROC()BYTE) RRCLSI;
   14  EXT PROC (PROC(BYTE)) RRCLSO;
   15
   16  EXT PROC(REAL)RWRT;
   17  EXT PROC()REAL RREAD;
   18  EXT PROC (INT) IWRT;
   19  EXT PROC(INT,INT)IWRTF;
   20  EXT PROC (REF INT)INT NPRIME;
   21  EXT PROC (REAL)REAL ZZZALG;
   22
   23
   24  EXT PROC (REF ARRAY REAL,REF ARRAY REAL) SINEG;
   25  ENT PROC RRJOB ();
   26     INT I:=2;
   27     PROC()BYTE TYIN:=RROPI(1);
   28     PROC(BYTE) TYOUT:=RROPOS(1),LPOUT:=RROPO(2);
   29     IN:=TYIN;
   30     OUT:=LPOUT;
   31     OUT(FF);
   32     TWRT("#NL#PRIME NUMBERS#NL#*************#NL#");
   33     TO 20 DO
   34        TO 5 DO
   35           I:=NPRIME(I);
   36           IWRTF(I,10);
   37           I:=I+1;
   38        REP;
   39        OUT(NL);
   40     REP;
   41     OUT(FF);
   42     TWRT("SINEWAVE#NL#********#NL#");
   43     OUT:=TYOUT;
   44     TWRT("#NL#INTERVAL=#VT#");
   45     OUT:=LPOUT;
   46     SINEPLOT();
   47  OUT:=TYOUT;
   48     BLOCK REAL X:=0.0;
   49        GOTO FIRST;
   50        WHILE X>0.0 DO
   51           OUT(NL);
   52           TWRT("#NL#LOGE (X)=");
   53           RWRT(ZZZALG(X));OUT(VT);
   54
   55  FIRST:
```

```
56              TWRT("#NL#X=#VT#");
57              X:=RREAD();
58           REP;
59        ENDBLOCK;
60
61
62        RRCLSI(TYIN);
63        RRCLSO(TYOUT);
64        RRCLSO(LPOUT);
65     ENDPROC;
66
67
68     DATA SINED;
69        ARRAY(100)REAL R1,R2;
70
71     ENDDATA;
72
73     PROC SINEPLOT();
74        REAL I:=RREAD();
75        FOR J:=1 TO 100 DO
76           R1(J):=IF J=1 THEN 0.0 ELSE R1(J-1)+I END;
77        REP;
78        SINEG(R1,R2);
79        FOR J:=1 TO 100 DO
80           INT K:=INT(39.0*R2(J));
81           OUT(10);
82           IF K >0 THEN
83              TO 39 DO OUT(' ') REP;
84              OUT('I');
85              TO K DO OUT('*') REP;
86           ELSEIF K=0 THEN
87              TO 39 DO OUT(' ') REP;
88              OUT('*');
89           ELSE
90              TO 39+K DO OUT(' ') REP;
91              TO -K DO OUT('*') REP;
92              OUT('I');
93           END;
94        REP;
95        OUT(NL);
96     ENDPROC;
97
98     :EOF

F 201 LINE 29
F 201 LINE 30
F 201 LINE 43
F 201 LINE 45
F 201 LINE 47
```

```
FORTRAN SUBPROGRAM REPORT

FORTRAN NAME "NPRIME"  PROCESSED LINE     35
FORTRAN NAME "ZZZALG"  PROCESSED LINE     53
FORTRAN NAME "SINEG "  PROCESSED LINE     78
COMPILATION OK
PROG ELAPSED TIME:  00H-01M-06S-060MS-
```

```
 HSFC VER.  02

      0000        0       IDENT ZZZALG
      0001        1             FUNCTION ZZZALG(R)
      0002        2             ZZZALG=ALOG(R)
      0003        3             RETURN
      0004        4             END
:EOF
PROG ELAPSED TIME:  00H-00M-05S-760MS-
```

```
 HSFC VER.  02

      0000        0       IDENT SINEG
      0001        1             SUBROUTINE SINEG(RAIN,RAOUT)
      0002        2             DIMENSION RAIN(100),RAOUT(100)
      0003        3             DO 10 I=1,100
      0004        4          10 RAOUT(I)=SIN(RAIN(I))
      0005        5             RETURN
      0006        6             END
:EOF
PROG ELAPSED TIME:  00H-00M-06S-080MS-
```

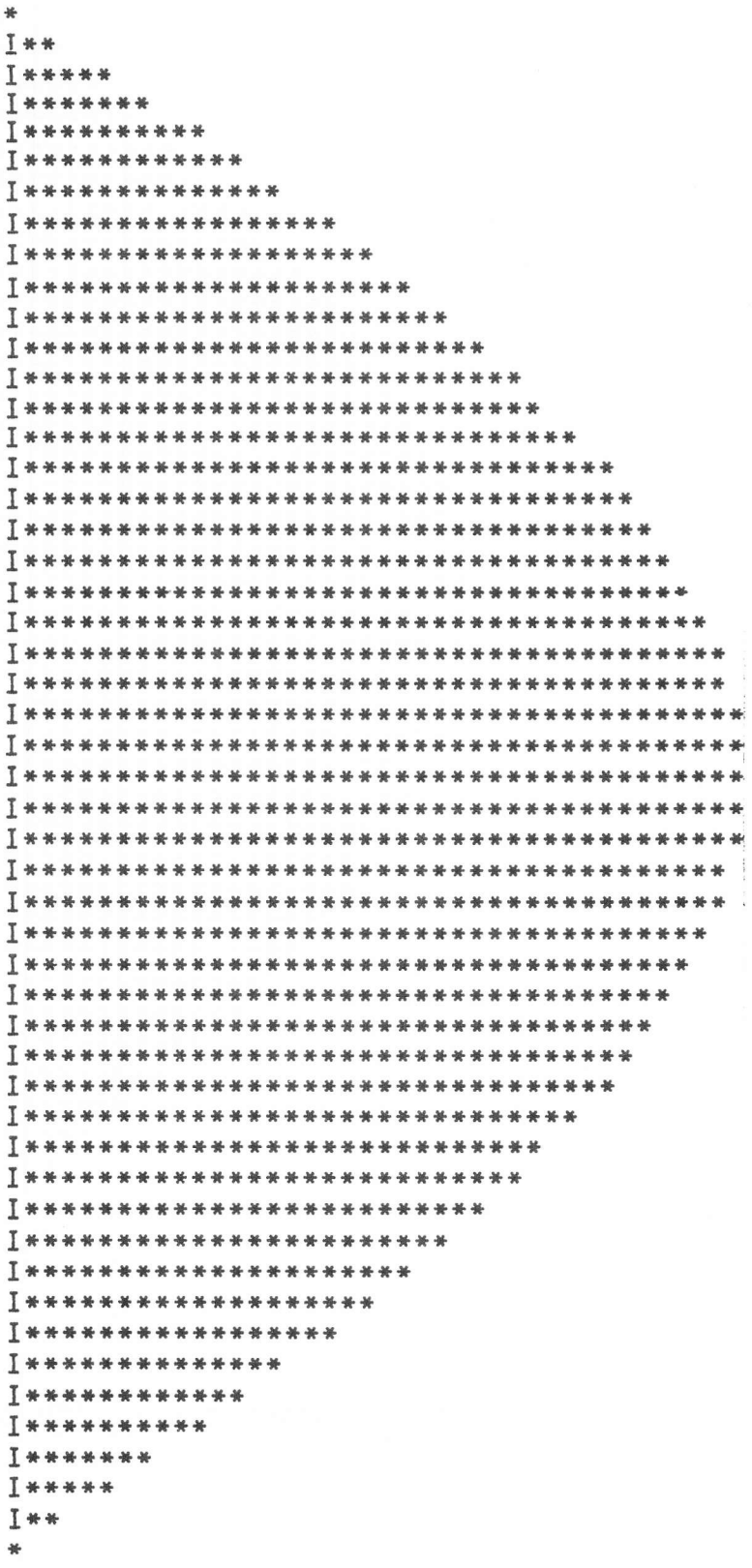
```
 HSFC VER.  02

      0000        0       IDENT NPRIME
      0001        1             FUNCTION NPRIME(I)
      0002        2             IL=I
      0003        3          10 R=IL
      0004        4             S=SQRT(R)
      0005        5             II=IFIX(S)+1
      0006        6             DO 20 J=2,II
      0007        7             IF(IL-((IL/J)*J))20,40,20
      0008        8          20 CONTINUE
      0009        9          30 NPRIME=IL
      0010       10             RETURN
      0011       11          40 IL=IL+1
      0012       12             GO TO 10
      0013       13             END
      0014
:EOF
PROG ELAPSED TIME:  00H-00M-08S-460MS-
```

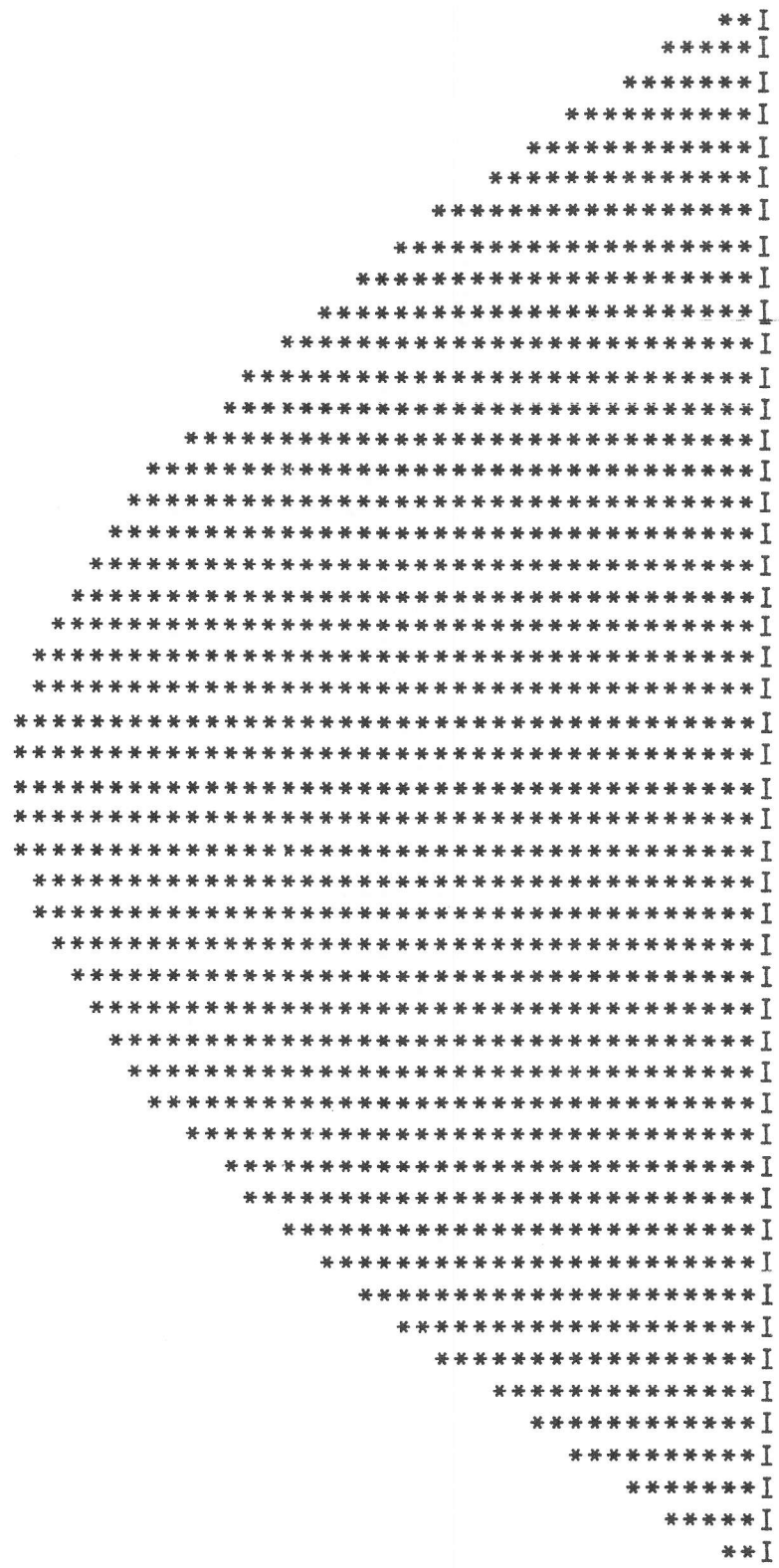
```
                    ***    SYMBOL    TABLE   ***

$FORT   2004 R    $NPRIME219E R    $SINEG 2090 R    %FORT   2002 R
%NPRIME2190 R     %SINEG 208A R    F:CLOR 2326 R    F:DL    23DE R
F:ER    22E6 R    F:ERR3 22D4 R    F:ERR4 22D8 R    F:ERR0 22DC R
F:ERR1 22E0 R     F:ERLK 22E4 R    F:FCT   **** U   F:FL    286C R
F:FX    27E4 R    F:ID    233A R   F:IM    2328 R   F:IOCV 2324 R
F:IX    2350 R    F:LPFC 0002 A    F:PA    227E R   F:RCOS 246E R
F:RD    2894 R    F:RECV 2322 R    F:RLOG 25C8 R    F:RM    29BC R
F:RN    2B8A R    F:RP    2ADC R   F:RS    2AE0 R   F:RSEV 2320 R
F:RSIN 2468 R     F:RSQR 2716 R    F:S2    23DA R   F:S4    23D6 R
F:S6    238E R    F:ST    2252 R   F:T1    27E8 R   F:TYFC 0001 A
INPROC 17BC R     IWRT    0DD6 R   IWRTF   0988 R   NPRIME 20BE R
R:R00   21AE R    R:R01   1184 R   R:R02   11B4 R   R:R03   0E48 R
R:R04   0E60 R    R:R05   0E6A R   R:R06   0E6A R   R:R07   0E7A R
R:R09   0E9A R    R:R10   0EB4 R   R:R12   0EBE R   R:R13   0EBE R
R:R14   0EFC R    R:R15   0F70 R   R:R16   0F86 R   R:R17   1056 R
R:R18   10F4 R    R:R19   10F8 R   R:R20   1100 R   R:R21   10FC R
R:R22   1104 R    R:R23   111A R   R:R24   1118 R   R:R25   113E R
R:R26   1154 R    R:R27   1150 R   R:R28   1146 R   RRACTV 000C A
RRASG   0017 A    RRATDE 000E A    RRCANR 001A A    RRCLSI 1E18 R
RRCLSO 1E6C R     RRCNAB 0007 A    RRCNLE 0014 A    RRCNTI 000A A
RRDCLE 0015 A     RRDCTI 000B A    RRDEL   0018 A   RRDTDE 000F A
RREAD   0B82 R    RRERR   000E A   RRERRX 0018 A    RREVEN 0012 A
RREXIT 0003 A     RRGBF   21C2 R   RRGEL   12F6 R   RRIOLK 0001 A
RRIPF   14D6 R    RRJOB   051C R   RRLDSE 0009 A    RRNUL   1508 R
RROPF   14F0 R    RROPI   1586 R   RROPMS 0019 A    RROPO   15CA R
RROPOS 1606 R     RRORDI 1CE6 R    RRORDO 1D82 R    RRPAUS 0006 A
RRRBF   21F6 R    RRSED   0006 A   RRSIO   0002 A   RRSTK   0008 A
RRSTR   142A R    RRSWIT 000D A    RRTIME 0011 A    RRWAIT 0002 A
RRWTIM 0016 A     RWRT    0A5A R   SINEG   201A R   TWRT    0A02 R
ZZZALG 1FCA R

                              ******

                    START = 142A   LENGTH = 2BAE   REGION = 1617
ERR.LKE.
:EOF
PROG ELAPSED TIME:  00H-02M-17S-040MS-
```

PRIME NUMBERS
****

3	5	7	11	13
17	19	23	29	31
37	41	43	47	53
59	61	67	71	73
79	83	89	97	101
103	107	109	113	127
131	137	139	149	151
157	163	167	173	179
181	191	193	197	199
211	223	227	229	233
239	241	251	257	263
269	271	277	281	283
293	307	311	313	317
331	337	347	349	353
359	367	373	379	383
389	397	401	409	419
421	431	433	439	443
449	457	461	463	467
479	487	491	499	503
509	521	523	541	547

SINEWAVE
****





PROG ELAPSED TIME: 00H-01M-04S-740MS-