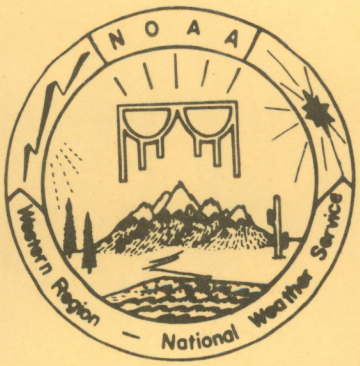
Western Region Computer Programs and
ems NWS WRCP - No. 3

DECODER FOR SIGNIFICANT LEVEL TRANSMISSIONS OF RAOBS

John A. Jannuzzi

National Weather Service Western Region
Salt Lake City, Utah
August 1979

noaa NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION / National Weather Service



PREFACE

This Western Region publication series is considered as a subset of our Technical Memorandum series. This series will be devoted exclusively to the exchange of information on and documentation of computer programs and related subjects. This series was initiated because it did not seem appropriate to publish computer program papers as Technical Memoranda; yet, we wanted to share this type of information with all Western Region forecasters in a systematic way. Another reason was our concern that in the developing AFOS-era there will be unnecessary and wasteful duplication of effort in writing computer programs in National Weather Service (NWS). Documentation and exchange of ideas and programs envisioned in this series hopefully will reduce such duplication. We also believe that by publishing the programming work of our forecasters, we will stimulate others to use these programs or develop their own programs to take advantage of the computing capabilities AFOS makes available.

We solicit computer-oriented papers and computer programs from forecasters for us to publish in this series. Simple and short programs should not be prejudged as unsuitable.

The great potential of the AFOS-era is strongly related to local computer facilities permitting meteorologists to practice in a more scientific environment. It is our hope that this new series will help in developing this potential into reality.

NOAA Western Region Computer Programs and Problems NWS WRCP

1 Standardized Format for Computer Series. June 1979.
2 AFOS Crop and Soil Information Report Program. Ken Mielke, July 1979.

NOAA Western Region Computer Programs and Problems NWS WRCP - No. 3

DECODER FOR SIGNIFICANT LEVEL TRANSMISSIONS OF RAOBS

John A. Jannuzzi
Scientific Services Division
Salt Lake City, Utah
August 1979

UNITED STATES
DEPARTMENT OF COMMERCE
Juanita M. Kreps, Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION
Richard A. Frank, Administrator

National Weather
Service
Richard E. Hallgren, Director

CONTENTS

DECODER FOR SIGNIFICANT LEVEL TRANSMISSIONS OF RAOBS

John A. Jannuzzi
Scientific Services Division
Salt Lake City, Utah

I. GENERAL INFORMATION

A. Summary:

This program decodes the significant level portion of a raob, which has been saved from an AFOS file to an RDOS file, and places the decoded information in two RDOS files; one containing temperature and dew-point information and the other containing the vertical-wind profile information. The decoded information is in a form easily accessed by other applications programs. Much of the program involves string searches for various code groups of the raob transmission.

B. Environment:

The program is written in Data General Fortran IV and is designed to run on the standard Eclipse S-230 minicomputer of AFOS. The program needs less than 10K memory to be executed; hence, it can run in the background partition (Background) of any WSO or WSFO while AFOS is running in the foreground partition (Foreground).

C. References:

This program was written to decode the significant level raob transmission as documented in Federal Meteorological Handbook No. 3, "Radiosonde Observations". This is not a complete decoder as some error information reported in the transmission is not used. The program has been used for two months prior to this publication so that major "bugs" could be discovered and corrected.

II. APPLICATION

A. Complete Program Description:

Before execution of this program, it is required that the raob message be SAVED as an RDOS file named "RAOB.BB". The program could have been written to access the raob directly as an AFOS file; thus, eliminating this step. This was not done, however, as little documentation was available to this programmer concerning AFOS file structure, but much was available on RDOS file structure. Likewise, the output files were created as RDOS files for later use by other programs.

The program begins by restructuring the file "RAOB.BB". Since each Data General Computer Word (two bytes) holds two ASCII characters (two single digit numbers or two letters), one has to look at each half of a word to examine individual characters. This can get awkward and messy when one doesn't know which half of the word (which byte) contains the desired character. To avoid this, the program splits the 512 words (2 blocks) of the array "IARRAY", which is nothing more than the core image of the "RAOB.BB" file, into 1024 words in an array called "IDATA"; each word containing the character in the right half and a null in the left half. The program then performs character or string searches through the array "IDATA" to find beginning points for decoding. The variable "N" is the counter used in the program to indicate which element of the array "IDATA" is being examined. The first search is for a four character string of "B" "B" "Space" "Space", for the beginning of the coded information (this gets the program past any AFOS headings). It then searches for the next 5-digit group, which is the date/time group and assigns the date, time, and station number. Next it searches for the "ØØ" group (surface data) and assigns the temperature and dew point there. Then the program enters an implied loop, which first searches for a "11" group. It assigns that group (pressure, temperature, and dew point). A check is then performed for a "22" group or end of message character ("=" or "51515" group). If a "22" group is found, it recycles through the loop to perform the next level assignments. This continues for "33", "44", etc., until the message is decoded.

When the temperature/dew-point part is decoded, it proceeds similarly into the wind portion of the message. The wind portion searches for "9" groups, assigning height levels, then assigning the corresponding wind information. When the next "9" group is found, it cycles through the procedure again (another implied loop).

The program ends by outputting this decoded information. Any previous files named "SOUNDING.T" and "SOUNDING.W" are deleted, new ones are created, and the information is output to these files. If at any time during the decode phase of the program, a code group is not found within the next few elements of "IDATA" (the exact number varies, but is usually 10 characters or words) or an obvious encoding error in the raob is discovered, the program will type out the message "THE DATA DOES NOT CONFORM TO FORMAT SPECIFICATIONS" on the dasher. The "N" number will also be printed on the dasher, telling the operator at what element of "IDATA" it gave up searching for the desired character. If this occurs in the temperature/dew-point portion of the program, it will jump to the wind portion of the program and continue on. If it occurs in the wind portion, the program will jump to the output portion.

B. Machine Requirements:

The program can be run in 10K memory; hence, in Background of any WSO or WSFO Eclipse computer while AFOS is running in Foreground. The executable save file requires 13824 bytes of storage (or about 27 blocks) on a disk. The three RDOS files that it requires for use, one for the sounding input and 2 for the program's output, have a combined disk requirement of about 1750 bytes (or about 3 blocks). Due to the disk accesses required by the program for input and output of data, and the faster read and write speeds of the rigid disk drives over the floppy disk drive, the program executes much faster from a rigid disk. Typical execution times in Background (while Foreground is also running) is about 17 seconds from a rigid disk and about 24 seconds from a floppy disk. The program assigns channel 6 as the input channel for the significant level raob transmission, channel 7 as the output channel for the temperature/dew-point information and channel 8 as the output channel for the wind information.

C. Structure of Software:

No subroutines are used. However, if a generalized string decoder were available, much of this could be simplified by making numerous calls to the string decoder. The first part of the program decodes the "TTBB" portion of the raob, the second part decodes the "PPBB" portion.

D. Data Base:

The input file and two output files need to reside in the same directory as the save file for this program; no directory specifier is used with any file or channel number assignment. The input file (channel 6) is entitled "RAOB.BB". This file is created by doing a SAVE of the significant level raob transmission into the RDOS file "RAOB.BB" (i.e., SAVE:SLCSGLSLC DPØ:RAOB.BB). The output files are created by the program itself. If the output filenames already exist, the old files are deleted and new ones are created. The temperature/dew-point file is named "SOUNDING.T". The wind file is named "SOUNDING.W".

III. PROCEDURES

A. Initiation of Program/B. Input Requirements:

The program requires that the significant level part of the raob be SAVED in the RDOS filename "RAOB.BB". On completing this, the program may be executed via the AFOS ADM. An example of these two steps is:

1) SAVE:SLCSGLSLC DPØ:RAOB.BB

2) RUN:DPØ:SGLDECODER.SV

C. Output:

The two output files "SOUNDING.T" and "SOUNDING.W" are in the same directory as the files "RAOB.BB" and "SGLDECODER.SV". These can be examined by typing the file on the dasher (i.e., TYPE DPØ:SOUNDING.T) or displaying it at an AFOS ADM (i.e., DSP:DPØ:SOUNDING.T). The two output files are created via standard Fortran formatted write statements.

The following is an example of a second transmission of a raob SAVED in RAOB.BB and the resultant two output files from this program.

```
SATSGLDRT          EWOUSOO KWRH 061636
 DRT 72260 TTBB  5612/ 72260 00974 21015 11960 24014 22850 19056
33800 17264 44700 10271 55576 00968 66565 01958 77551 02967
88487 08164 99472 10556 11455 11563 22400 20558 33373 24163
44275 38780 55176 585// 66106 699// 77100 669//=
PPBB  56120 72260 90023 13003 21518 20015 90467 19013 16509
16011 9089/ 17012 15511 91024 13012 10012 10514 91679 10015
08514 10015 92056 07509 15509 14511 928// 09511 93034 05006
07511 08513 9357/ 05516 08524 94249 04020 06027 05023 9504/
05522 09015=
```

```
TYPE DP1:SOUNDING.W
260  6 12
 0 130 3
 2 21518
 3 20015
 4 19013
 6 165 9
 7 16011
 8 17012
 9 15511
10 13012
12 10012
14 10514
16 10015
17  8514
19 10015
20  75 9
25 155 9
26 14511
28  9511
30  50 6
33  7511
34  8513
35  5516
37  8524
42  4020
44  6027
49  5023
50  5522
54  9015
```

```
TYPE DP1:SOUNDING.T
260   6 12
 974   21.0   1.5
 960   24.0   1.4
 850   19.0   6.0
 800   17.2  14.0
 700   10.2  21.0
 576   -0.9  18.0
 565   -1.9   8.0
 551   -2.9  17.0
 487   -8.1  14.0
 472  -10.5   6.0
 455  -11.5  13.0
 400  -20.5   8.0
 373  -24.1  13.0
 275  -38.7  30.0
 176  -58.5  -1.0
 106  -69.9  -1.0
 100  -66.9  -1.0
```

The wind output lists the height in thousands of feet and the wind direction and speed. The temperature output lists the pressure level, the temperature, and the dew-point depression (a depression of -1.0 indicates no depression reported).

D. Cautions and Restrictions:

The program will only decode a RAOB.BB file that is at most 2 blocks in length. I cannot foresee a raob longer than this, however. If the raob transmission is garbled, code groups are missing, etc., the program will type out the message "THE DATA DOES NOT CONFORM TO FORMAT SPECIFICATIONS" on the dasher, and that section of the decode (TTBB or PPBB) execution will be aborted. Also, with that message will be a number (N) which is the byte location number where the program began having problems.

E. Complete Program Listing:

(See following pages.)

```
C     DECODES THE TTBB RAOB TRANSMITION INTO TWO RDOS FILES:
C     ONE FOR TEMPERATURES AND DEWPOINTS, AND ONE FOR WINDS.  IF
C     THE FORMAT IN THE FILE IS NOT CORRECT, IT WILL TYPE OUT A
C     FLAG MESSAGE TO THAT EFFECT ON THE DASHER AND EXIT THAT
C     PORTION OF THE PROGRAM.
C
C     SPLITS THE "WORDS" OF THE AFOS FILE NOW STORED IN "RAOB.BB"
C     INTO THE TWO BYTE COMPONENTS AND PUTS EACH BYTE ASCII CHARACTER
C     INTO SEPARATE WORDS OF A NEW ARRAY NAMED "IARRAY".
      DIMENSION IARRAY(0:511),IDATA(1024),IPRESS(50),TEMP(50),DEPR(50),
     1ILVL(50),IDIR(50),ISPEED(50)
      CALL OPEN(6,"RAOB.BB",2,IER)
      CALL WRBLK(6,0,IARRAY,2,IER)
      IMASK1=177400K
      IMASK2=377K
      DO 25 I=0,511
      N=(I+1)*2
      NM1=(I*2)+1
      IDATA(N)=IAND(IMASK2,IARRAY(I))
      IDATA(NM1)=IAND(IMASK1,IARRAY(I))
      IDATA(NM1)=ISHFT(IDATA(NM1),-8)
   25 CONTINUE
C     N IS A CHARACTER POINTER USED TO GUIDE US THRU THE ARRAY "IDATA"
      N=1
C     STRING SEARCH FOR "B""B""SPACE""SPACE"
      DO 50 I=1,70
      IF(IDATA(I) .NE. 102K) GO TO 49
      IP1=I+1
      IF(IDATA(IP1) .NE. 102K) GO TO 49
      IP2=I+2
      IF(IDATA(IP2) .NE. 040K) GO TO 49
      IP3=I+3
      IF(IDATA(IP3) .NE. 040K) GO TO 49
      N=I+4
      GO TO 51
   49 N=N+1
   50 CONTINUE
      WRITE(10,3) N
    3 FORMAT(5X,I3,1X,"DATA DOES NOT CONFORM TO FORMAT SPECIFICATIONS")
      GO TO 215
C     ASSIGNS THE DATE, TIME, AND STATION NUMBER.
   51 NP1=N+1
      NP2=N+2
      NP3=N+3
      IDATE=(10*(IDATA(N)-48))+(IDATA(NP1)-48)-50
      ITIME=(10*(IDATA(NP2)-48))+(IDATA(NP3)-48)
      N=N+8
      NP1=N+1
      NP2=N+2
      ISTATN=(100*(IDATA(N)-48))+(10*(IDATA(NP1)-48))+(IDATA(NP2)-48)
      N=N+4
      M=1
      IPOINT=0
      NP20=N+20
```

```
C       STRING SEARCH FOR "00" GROUP.
        DO 70 I=N,NP20
        IP1=I+1
        IF((IDATA(I)-48) .EQ. IPOINT .AND. (IDATA(IP1)-48) .EQ. IPOINT)
       1GO TO 80
        N=N+1
   70 CONTINUE
        WRITE(10,3) N
        GO TO 215
C       ASSIGNS THE PRESSURE AT THIS LEVEL.
   80 N=N+2
        NP1=N+1
        NP2=N+2
        IF(IDATA(N) .EQ. 057K) M=M-1
        IF(IDATA(N) .EQ. 057K) GO TO 73
        IPRESS(M)=(100*(IDATA(N)-48))+(10*(IDATA(NP1)-48))+(IDATA(NP2)-48)
C       STRING SEARCH FOR THE FIRST INTEGER; 0 THRU 9.
   73 N=N+3
        NP10=N+10
        DO 75 I=N,NP10
        DO 76 IJ=0,9
        IF(IDATA(I) .EQ. 057K) N=N+3
        IF(IDATA(I) .EQ. 057K) GO TO 91
        IF((IDATA(I)-48) .EQ. IJ) GO TO 85
   76 CONTINUE
        N=N+1
   75 CONTINUE
        WRITE(10,3) N
        GO TO 215
C       ASSIGNS THE TEMPERATURE AND DEW POINT DEPRESSION AT THIS LEVEL.
   85 NP1=N+1
        NP2=N+2
        ITEMP=((IDATA(N)-48)*100)+((IDATA(NP1)-48)*10)+(IDATA(NP2)-48)
        IISIGN=(ITEMP/2)*2
        IF(ITEMP .NE. IISIGN) ITEMP=-ITEMP
        TEMP(M)=(FLOAT(ITEMP)/10.0)
        N=N+3
        NP1=N+1
        IF(IDATA(N) .EQ. 057K) DEPR(M)=-1.0
        IF(IDATA(N) .EQ. 057K) GO TO 91
        IDEPR=(10*(IDATA(N)-48))+(IDATA(NP1)-48)
        DEPR(M)=FLOAT(IDEPR)
        IF(DEPR(M) .LE. 50) DEPR(M)=(DEPR(M)/10.0)
        IF(DEPR(M) .GE. 55.0) DEPR(M)=DEPR(M)-50.0
C       CHECK FOR AN "=" SIGN DENOTING THE END OF THE RAOB.
   91 N=N+2
        IF(IDATA(N) .EQ. 075K) GO TO 215
C       STRING SEARCH FOR THE FIRST INTEGER; 0 THRU 9.
        NP10=N+10
        DO 93 I=N,NP10
        DO 94 IJ=0,9
        IF((IDATA(I)-48) .EQ. IJ) GO TO 95
   94 CONTINUE
        N=N+1
   93 CONTINUE
```

```
      WRITE(10,3) N
      GO TO 215
C     CHECK TO SEE IF THIS IS THE NEXT LEVEL GROUP OR IF IT IS
C     THE END OF THE "TTBB" SECTION OF THE RAOB; THE 51515 GROUP.
   95 IPOINT=IPOINT+1
      IF(IPOINT .EQ. 10) IPOINT=1
      NP1=N+1
      NP2=N+2
      NP3=N+3
      NP4=N+4
      IF(IDATA(N) .EQ. 065K .AND. IDATA(NP1) .EQ. 061K .AND.
     1IDATA(NP2) .EQ. 065K .AND. IDATA(NP3) .EQ. 061K .AND. IDATA(NP4)
     1.EQ. 065K) GO TO 215
      IF((IDATA(N)-48) .NE. IPOINT .AND. (IDATA(NP1)-48) .NE. IPOINT)
     1WRITE(10,3) N
      IF((IDATA(N)-48) .NE. IPOINT .AND. (IDATA(NP1)-48) .NE. IPOINT)
     1GO TO 215
      M=M+1
C     RECYCLES THRU THE LOOP TO ASSIGN VALUES AT THE NEW PRESSURE LEVEL.
      GO TO 80
C     STRING SEARCH FOR THE WIND SECTION OF THE RAOB; "B""B"
C     "SPACE""SPACE".
  215 ISAVE=M
      NP20=N+20
      DO 200 I=N,NP20
      IF(IDATA(I) .NE. 102K) GO TO 206
      IP1=I+1
      IF(IDATA(IP1) .NE. 102K) GO TO 206
      IP2=I+2
      IF(IDATA(IP2) .NE. 040K) GO TO 206
      IP3=I+3
      IF(IDATA(IP3) .NE. 040K) GO TO 206
      N=N+4
      GO TO 201
  206 N=N+1
  200 CONTINUE
      WRITE(10,3) N
      GO TO 300
C     STRING SEARCH FOR THE BEGINNING OF THE FIRST GROUP SPECIFYING
C     THE WIND LEVELS; A "SPACE""9".
  201 M=1
      NP10=N+50
      DO 205 I=N,NP10
      IP1=I+1
      IF(IDATA(I) .NE. 040K) GO TO 208
      IF((IDATA(IP1)-48) .NE. 9) GO TO 208
      GO TO 207
  208 N=N+1
  205 CONTINUE
      WRITE(10,3) N
      GO TO 300
```

```
C     ASSIGNS THE NEXT LEVEL OF WINDS GIVEN (IT CHECKS FOR "/"
C     FOR MISSING LEVELS).
  207 N=N+2
      NP1=N+1
      NP3=N+3
      MM=M
      DO 210 I=NP1,NP3
      IF(IDATA(I) .NE. 057K) ILVL(M)=(10*(IDATA(N)-48))+(IDATA(I)-48)
      IF(IDATA(I) .NE. 057K) M=M+1
  210 CONTINUE
C     ASSIGNS THE WINDS AT THE ABOVE DETERMINED LEVELS.
      N=N+4
      M=M-1
      DO 220 II=MM,M
      NP10=N+10
      DO 211 I=N,NP10
      DO 212 IJ=0,9
      IF((IDATA(I)-48) .EQ. IJ) GO TO 216
  212 CONTINUE
      N=N+1
  211 CONTINUE
      WRITE(10,3) N
      GO TO 300
  216 NP1=N+1
      NP2=N+2
      NP3=N+3
      NP4=N+4
      IDIR(II)=((IDATA(N)-48)*100)+((IDATA(NP1)-48)*10)+(IDATA(NP2)-48)
      ISPEED(II)=((IDATA(NP3)-48)*10)+(IDATA(NP4)-48)
      N=N+5
  220 CONTINUE
C     CHECKS FOR END OF FILE; "=".
      IF(IDATA(N) .EQ. 075K) GO TO 300
C     CHECK FOR NEXT "9" GROUP AND MAKES SURE THAT ALL THE
C     CHARACTERS IN THE GROUP ARE INTEGERS OR A "/".
      NP10=N+10
      DO 230 I=N,NP10
      IF((IDATA(I)-48) .NE. 9) GO TO 229
      DO 232 IIJ=1,4
      DO 231 IJ=0,10
      JI=IIJ+I
      IF((IDATA(JI)-48) .EQ. IJ .OR. IDATA(JI) .EQ. 057K)
     1GO TO 232
      IF(IJ .EQ. 10) WRITE(10,3) N
      IF(IJ .EQ. 10) GO TO 300
  231 CONTINUE
  232 CONTINUE
      GO TO 235
  229 N=N+1
  230 CONTINUE
C     RECYCLES THRU THE LOOP TO ASSIGN THE WINDS AT THE NEXT LEVELS.
  235 M=M+1
      N=N-1
      GO TO 207
```

```
C     WRITES THE DECODED TEMPERATURE INFORMATION INTO AN RDOS FILE
C     NAMED "SOUNDING.T" AND THE DECODED WIND INFORMATION INTO AN RDOS
C     FILE NAMED "SOUNDING.W".
  300 CALL DFILW("SOUNDING.T",IER)
      CALL CFILW("SOUNDING.T",2,IER)
      CALL DFILW("SOUNDING.W",IER)
      CALL CFILW("SOUNDING.W",2,IER)
      CALL OPEN(7,"SOUNDING.T",2,IER)
      WRITE(7,304) ISTATN,IDATE,ITIME
  304 FORMAT(1X,I3,1X,I2,1X,I2)
      DO 310 I=1,ISAVE
      IF(IPRESS(I) .LT. 100) IPRESS(I)=IPRESS(I) + 1000
      WRITE(7,305) IPRESS(I), TEMP(I), DEPR(I)
  305 FORMAT(1X,I4,1X,F5.1,1X,F4.1)
  310 CONTINUE
      CALL OPEN(8,"SOUNDING.W",2,IER)
      WRITE(8,304)ISTATN,IDATE,ITIME
      DO 320 I=1,M
      WRITE(8,325) ILVL(I), IDIR(I), ISPEED(I)
  325 FORMAT(1X,I2,1X,I3,I2)
  320 CONTINUE
      CALL CLOSE(6,IER)
      CALL CLOSE(7,IER)
      CALL CLOSE(8,IER)
      STOP
      END
R
```

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

NOAA, the *National Oceanic and Atmospheric Administration*, was established as part of the Department of Commerce on October 3, 1970. The mission responsibilities of NOAA are to monitor and predict the state of the solid Earth, the oceans and their living resources, the atmosphere, and the space environment of the Earth, and to assess the socioeconomic impact of natural and technological changes in the environment.

The six Major Line Components of NOAA regularly produce various types of scientific and technical information in the following kinds of publications:

PROFESSIONAL PAPERS — Important definitive research results, major techniques, and special investigations.

TECHNICAL REPORTS—Journal quality with extensive details, mathematical developments, or data listings.

TECHNICAL MEMORANDUMS — Reports of preliminary, partial, or negative research or technology results, interim instructions, and the like.

CONTRACT AND GRANT REPORTS—Reports prepared by contractors or grantees under NOAA sponsorship.

TECHNICAL SERVICE PUBLICATIONS—These are publications containing data, observations, instructions, etc. A partial listing: Data serials; Prediction and outlook periodicals; Technical manuals, training papers, planning reports, and information serials; and Miscellaneous technical publications.

ATLAS—Analysed data generally presented in the form of maps showing distribution of rainfall, chemical and physical conditions of oceans and atmosphere, distribution of fishes and marine mammals, ionospheric conditions, etc.

Information on availability of NOAA publications can be obtained from:

ENVIRONMENTAL SCIENCE INFORMATION CENTER
ENVIRONMENTAL DATA SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

3300 Whitehaven Street, N.W.
Washington, D.C. 20235