A Eastern Region Computer Programs
Problems NWS ERCP - No. 13

A DECODER FOR MANUALLY DIGITIZED RADAR OBSERVATIONS

Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, Ohio

Scientific Services Division
Eastern Region Headquarters
June 1983

U.S. DEPARTMENT OF COMMERCE / National Oceanic and Atmospheric Administration / National Weather Service

NOAA Technical Memorandum
National Weather Service, Eastern Region Computer Programs and Problems

The Eastern Region Computer Programs and Problems (ERCP) series is a subset of the Eastern Region Technical Memorandum series. It will serve as the vehicle for the transfer of information about fully documented AFOS application programs. The format of ERCP - No. 1 will serve as the model for future issuances in this series.

1 An AFOS version of the Flash Flood Checklist. Cynthia M. Scott, March 1981. (PB81 211252).

2 An AFOS Applications Program to Compute Three-Hourly Stream Stages. Alan P. Blackburn, September 1981. (PB82 156886).

3 PUPPY (AFOS Hydrologic Data Reporting Program). Daniel P. Provost, December 1981. (PB82 199720).

4 Special Search Computer Program. Alan P. Blackburn, April 1982. (PB83 175455).

5 Conversion of ALEMBIC$ Workbins. Alan P. Blackburn, October 1982. (PB83 138313).

6 Real-Time Quality Control of SAOs. John A. Billet, January 1983. (PB83 166082).

7 Automated Hourly Weather Collective from HRR Data Input. Lawrence Cedrone, January 1983. (PB83 167122).

8 Decoders for FRH, FTJ and FD Products. Cynthia M. Scott, February 1983. (PB83 176057).

9 Stability Analysis Program. Hugh M. Stone, March 1983. (PB83 197947).

10 Help for AFOS Message Comp. Alan P. Blackburn, May 1983.

11 Stability and Other Parameters from the First Transmission RAOB Data. Charles D. Little, May 1983.

12 TERR, PERR, and BIGC: Three Programs to Compute Verification Statistics. Matthew R. Peroutka, August 1983.

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NOAA
U.S. DEPARTMENT OF COMMERCE

NOAA Eastern Region Computer Programs and Problems - No. 13

A DECODER FOR MANUALLY DIGITIZED RADAR OBSERVATIONS

Matthew R. Peroutka
National Weather Service Forecast Office
Cleveland, Ohio

Scientific Services Division
Eastern Region Headquarters
June 1983

A DECODER FOR MANUALLY DIGITIZED RADAR OBSERVATIONS

Matthew R. Peroutka
WSFO, Cleveland, Ohio

I. General Information

A. Introduction

Most weather data is transmitted in some form of code. AFOS applications programmers have spent a great deal of time writing subroutines which can convert these coded messages into arrays or files which could then be easily accessed by other routines. This decoder fills a gap in that collection.

The routines were actually developed as part of a much larger project which will eventually convert radar observations to rainfall amounts.

Hourly radar reports transmitted through the AFOS system contain a manually digitized section. To encode such a report, the operator uses a gridded overlay and records the VIP levels of the strongest echo in each grid. Each grid box in the overlay is identified by two letters and it corresponds to a grid box on a national grid system.

To maintain a structured approach, the decoding has been divided among six FORTRAN subroutines. Although this paper is meant to describe the decoder's operation, it is worthwhile to point out a few of the modules. Function JDATE and subroutine J2MDA convert from month/day/year to Julian dates and back. Function MASK is a very powerful pattern-search algorithm. It will search any part of an unpacked string for a sub-string which matches a given pattern. The pattern mask can be adjusted for exact matches, ranges of values, or any match ("wild card").

B. Environment

These programs were developed for a Data General Eclipse using Data General's FORTRAN IV. The utility libraries BG.LB, UTIL.LB, and FORT.LB are needed as well as the AFOSE.LB system library.

C. References

National Weather Service Radar Code User's Guide, December 1980, National Weather Service, Silver Spring, Maryland

II. Program Reference

A. ROBDEC

1. Program description

ROBDEC will search the current and previous versions of the current product for an observation with a given date and time. The creation date-time polynomial must be within two hours of the requested date and time. The date in the WMO header must be the date (or the next date) requested, and the time in the body of the observation must agree with the time requested. If any of these fail to match, a previous version is searched.

If no problems are encountered, the decoded observation is then loaded into a thirteen-by-thirteen integer array. The array values correspond to the VIP levels reported in the MDR portion of the observation. Array element (7,7) corresponds to grid box MM.

Finally, the entire observation is searched for any of the six operational contractions. If PPINE is found, the array is set to zero.

2. Call statement and external variables

```
CALL ROBDEC(IDA,IHR,NVER,ITYP,MDRAR)
```

The variables are:

IDA--The date array. IDA(1) is the month, IDA(2) is the day, and IDA(3) is the year.

IHR--The hour. If the observation was taken at 1330Z, IHR would be 13.

NVER--The number of versions stored in the local database.

ITYP--Return code:

```
ITYP =  -1 for decode failure
         0 for success
         1 for PPINE
         2 for PPIOM
         3 for PPINA
         4 for ROBEPS
         5 for ARNO
         6 for RHINO.
```

MDRAR--The output array.

III. Program listings

on following pages

```
      SUBROUTINE ROBDEC(IDA,IHR,NVER,ITYP,MDRAR)
      DIMENSION IDA(3), MDRAR(13,13), IBF(128), IUP(512)
C
C      THIS SUBROUTINE SEARCHES VERSIONS OF THE CURRENT PRODUCT TO FIND A ROB
C      FOR THE DATE (IDA) AND TIME (IHR) REQUESTED.  THE MDR SECTION OF THE OB
C      IS USED TO FILL THE 13X13 ARRAY (MDRAR) WITH MDR VAULUES.  ITYP =
C
C      -1 FOR FAILURE   0 FOR NO PROBLEMS   1 FOR PPINE   2 FOR PPIOM
C       3 FOR PPINA     4 FOR ROBEPS        5 FOR ARNO    6 FOR RHINO.
C
      CALL GROB(IDA,IHR,NVER,IBF,IUP,I)              ;CHECK DATES AND TIMES.
      IF (I.EQ.-1) GOTO 500
      IEND = MASK("<0><203>",1,IUP,I,256)            ;SEARCH FOR END OF PRODUCT
      IF (IEND.NE.-1) GOTO 100                       ;IN FIRST BLOCK.
       CALL RDBKF(1,IBF,IER)                         ;LOAD SECOND BLOCK.
       IF (IER.NE.1) GOTO 500
       CALL UNPACK(IBF,256,IUP(257))
       IEND = MASK("<0><203>",1,IUP,257,512)
       IF (IEND.EQ.-1) IEND = 512
  100 CALL ROBOP(IUP,I,IEND,ITYP)                    ;SEARCH FOR OPERATIONAL
      IF (ITYP.EQ.2.OR.ITYP.EQ.3) GOTO 550           ;CONTRACTIONS.
      DO 200 I1 = 1, 13                              ;ZERO OUT THE ARRAY.
       DO 200 J1 = 1, 13
  200 MDRAR(I1,J1) = 0
      IF (ITYP.EQ.1) GOTO 550
      I = MASK("<0><136>",1,IUP,I,IEND)              ;SEARCH FOR ↑.
      IF (I.EQ.-1) GOTO 500
      IE = MASK("<0><75>",1,IUP,I,IEND)              ;SEARCH FOR =.
      IF (IE.NE.-1) IEND = IE
      I = I + 1
      IEND = IEND - 1
  300 I = MASK("GSGS09",3,IUP,I,IEND)                ;SEACH FOR THE PATTERN
      IF (I.EQ.-1) GOTO 550                          ;"LETTER-LETTER-NUMBER".
      I1 = IUP(I) - 70                               ;ANALYZE THE TWO LETTERS.
      J1 = IUP(I+1) - 70
      DO 400 J = 2, 14                               ;NOW LOOK AT THE CHARACTERS
       K = IUP(I+J) - 48                             ;FOLLOWING.
       IF (K.GE.0.AND.K.LE.9) GOTO 350               ;  IS IT A NUMBER?
        I = I + J                                    ;  NO.  SEARCH AGAIN.
        GOTO 300
  350  MDRAR(I1,J1+J-2) = K                          ;  YES.  LOAD IT INTO THE
  400 CONTINUE                                       ;  MDR ARRAY.
      I = I + 14
      GOTO 300
  500 ITYP = -1                                      ;FAILURE.
  550 RETURN
      END
      SUBROUTINE GROB(IDA,IHR,NVER,IBF,IUP,J)
      DIMENSION IDA(3), IBF(128), IUP(512)
C
C      THIS SUBROUTINE CHECKS ALL VERSIONS OF THE CURRENT PRODUCT TO FIND A
C      ROB WITH THE REQUESTED DATE AND TIME.  ON RETURN, J POINTS TO THE FIRST
C      DIGIT OF THE TIME IN IUP.  J = -1 INDICATES AN ERROR.
C
      JD = JDATE(IDA(1),IDA(2),IDA(3))
      CTIML = (JD-1)*1440. + (IHR-2)*60.          ;THE WINDOW IN JULIAN
      CTIMU = CTIML + 240.                        ;MINUTES.
      DO 500 I = 1, NVER                          ;LOOP TO CALL UP PREVIOUS
       CALL RDBKF(0,IBF,IER)                      ;VERSIONS.
       IF (IER.NE.1) GOTO 700
       CALL UNPACK(IBF,256,IUP)
       PTIM = IUP(17)*16384. + IUP(18)*128. + IUP(19)   ;CREATION TIME.
       IF (PTIM.LT.CTIML) GOTO 700                ;TOO OLD.  ERROR.
       IF (PTIM.GE.CTIMU) GOTO 400                ;TOO NEW.  TRY PRVS VERSN.
       J = MASK("<0> <0>K0Z0Z0Z<0> ",6,IUP,30,40)  ;SEARCH FOR " KCCC ".
       IF (J.EQ.-1) GOTO 400
       J = J + 6                                  ;J NOW POINTS TO DAY.
       IPD = (IUP(J) - 48)*10 + IUP(J+1) - 48
       IF (IPD.EQ.IDA(2)) GOTO 100                ;RIGHT DAY.
       JD = JD + 1                                ;CHECK FOR 00Z CROSSING.
       CALL J2MDA(JD,M,ND,IY)
       IF (ND.NE.IPD) GOTO 400
  100  J = J + 6                                  ;SEARCH FOR TIME IN ROB.
       J = MASK("<0> 09090909",5,IUP,J,J+20)
       IF (J.EQ.-1) GOTO 400
       J = J + 1                                  ;CHECK THIS TIME.
       IPH = (IUP(J) - 48)*10 + IUP(J+1) - 48
       IF (IPH.EQ.IHR) GOTO 750
  400  CALL PRVRF(IER)                            ;GET PREVIOUS VERSION.
       IF (IER.NE.1) GOTO 700
  500 CONTINUE
  700 J = -1
  750 RETURN
      END
      SUBROUTINE ROBOP(IUP,IBGN,IEND,ITYP)
      DIMENSION IUP(512)
C
C      THIS SUBROUTINE SEARCHES THE RADAR OBSERVATION UNPACKED IN IUP FROM
C      IBGN TO IEND FOR OPERATIONAL CONTRACTIONS.  ITYP IS SET TO:
C
C      0 FOR NO CONTRACTIONS   1 FOR PPINE   2 FOR PPIOM   3 FOR PPINA
C      4 FOR ROBEPS            5 FOR ARNO    6 FOR RHINO.
C
      ITYP = 1
      IF (MASK("<0>P<0>P<0>I<0>N<0>E",5,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 2
      IF (MASK("<0>P<0>P<0>I<0>O<0>M",5,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 3
      IF (MASK("<0>P<0>P<0>I<0>N<0>A",5,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 4
      IF (MASK("<0>R<0>O<0>B<0>E<0>P<0>S",6,IUP,IBGN,IEND).NE.-1)
     +        GOTO 100
      ITYP = 5
      IF (MASK("<0>A<0>R<0>N<0>O",4,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 6
      IF (MASK("<0>R<0>H<0>I<0>N<0>O",5,IUP,IBGN,IEND).NE.-1) GOTO 100
      ITYP = 0
  100 RETURN
      END
      FUNCTION MASK(MSK,LMSK,IUP,IBGN,ISTOP)
      DIMENSION MSK(1), IUP(1)
C
C      THIS FUNCTION SEARCHES IUP (AN UNPACKED ARRAY OF ASCII CHARACTERS) FROM
C      IBGN TO ISTOP FOR A STRING WHICH MATCHES THE MASK MSK.  MSK IS LMSK
C      WORDS IN LENGTH AND HAS THE FOLLOWING CHARACTERISTICS:
C
C           1.  IF THE LEFT BYTE OF A WORD IS 0, THE RIGHT BYTE NEEDS AN
C               EXACT MATCH.
C           2.  IF THE LEFT BYTE IS NON-ZERO, THEN A MATCH MUST BE GREATER
C               THAN OR EQUAL TO THE LEFT BYTE, AND LESS THEN OR EQUAL TO THE
C               RIGHT BYTE.  "09" OR "AZ", E. G.
C           3.  IF A WORD EQUALS -1, IT WILL MATCH ANY BYTE (A WILD CARD).
C
C      THE FUNCTION RETURNS THE LOCATION IN IUP OF THE BEGINNING OF THE
C      SUBSTRING.  MASK RETURNS -1 IF THE SEARCH FAILS.
C
      LIMIT = ISTOP - LMSK + 1
      DO 600 MASK = IBGN, LIMIT
       DO 500 I = 1, LMSK
        M = MSK(I)
        IF (M.EQ.-1) GOTO 500            ;WILD CARD.
        ML = ISHFT(M,-8)
        MR = IAND(M,377K)
        L = IUP(MASK+I-1)
        IF (ML.NE.0) GOTO 200            ;NEED AN EXACT MATCH.
         IF (MR.EQ.L) GOTO 500
         GOTO 600
  200   IF (L.LT.ML.OR.L.GT.MR) GOTO 600   ;RANGE FOR MATCH.
  500  CONTINUE
       GOTO 700
  600 CONTINUE
      MASK = -1                          ;UNSUCCESSFUL SEARCH.
  700 RETURN
      END
      INTEGER FUNCTION JDATE(MONTH,IDAY,IYEAR)
C
C      THIS FUNCTION WILL GENERATE A JULIAN DATE FOR THE MONTH, DAY, AND
C      YEAR INPUT.  IF THERE IS A PROBLEM WITH THE INPUT DATA, THE FUNC-
C      TION WILL RETURN A VALUE OF ZERO.
C
      COMMON /QJULQ/ MLIST(12)
      DATA MLIST/31,28,31,30,31,30,31,31,30,31,30,31/
      JDATE = 0                         ;CHECK INPUT DATA.
      IF (MONTH.LT.1.OR.MONTH.GT.12) RETURN
      IF (IYEAR.LT.1) RETURN
      IF (IDAY.LT.1) RETURN
      MLIST(2) = 28
      I = IYEAR/4*4                     ;IS THIS A LEAP YEAR?
      IF (I.EQ.IYEAR) MLIST(2) = 29
      I = IYEAR/100*100
      IF (I.EQ.IYEAR) MLIST(2) = 28
      I = IYEAR/400*400
      IF (I.EQ.IYEAR) MLIST(2) = 29
      IF (IDAY.GT.MLIST(MONTH)) RETURN
      DO 100 I = 1,MONTH                ;COMPUTE JULIAN DATE.
  100 JDATE = JDATE + MLIST(I)
      JDATE = JDATE - MLIST(MONTH) + IDAY
      RETURN
      END
      SUBROUTINE J2MDA(JDA,M,ID,IY)
      DIMENSION MS(12)
C
C      THIS SUBROUTINE ACCEPTS A JULIAN DATE (JDA) AND A YEAR (IY).  IT
C      PRODUCES A MONTH (M) AND A DAY (ID).  IF THE JULIAN DATE IS BAD,
C      M AND ID WILL BOTH EQUAL ZERO.
C
      M = 0
      ID = 0
      IF (JDA.LE.0) RETURN                 ;CHECK FOR BAD JDA.
      MS(1) = 31                           ;SET UP MONTH DATA.
      MS(2) = 28
      I = IY/4*4                           ;LEAP YEARS.
      IF (I.EQ.IY) MS(2) = 29
      I = IY/100*100
      IF (I.EQ.IY) MS(2) = 28
      I = IY/400*400
      IF (I.EQ.IY) MS(2) = 29
      MS(3) = 31
      MS(4) = 30
      MS(5) = 31
      MS(6) = 30
      MS(7) = 31
      MS(8) = 31
      MS(9) = 30
      MS(10) = 31
      MS(11) = 30
      MS(12) = 31
      J = 0                                ;SEARCH FOR MONTH.
      DO 100 I = 1,12
      M = I
      J = J + MS(I)
      IF (J.GE.JDA) GOTO 200
  100 CONTINUE
      M = 0                                ;JDA IS TOO BIG.
      RETURN
  200 IF (J.GT.JDA) GOTO 300
      ID = MS(M)
      RETURN
  300 ID = JDA + MS(M) - J
      RETURN
      END
```

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

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 assess the socioeconomic impact of natural and technological changes in the environment and 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.

The major 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.

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

ATLAS — Presentation of analyzed data generally 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.

TECHNICAL SERVICE PUBLICATIONS — Reports containing data, observations, instructions, etc. A partial listing includes data serials; prediction and outlook periodicals; technical manuals, training papers, planning reports, and information serials; and miscellaneous technical publications.

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.

Information on availability of NOAA publications can be obtained from:

ENVIRONMENTAL SCIENCE INFORMATION CENTER (D822)
ENVIRONMENTAL DATA AND INFORMATION SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

6009 Executive Boulevard
Rockville, MD 20852