National Aeronautics and Space Administration
Goddard Earth Science
Data Information and Services Center (GES DISC)

README Document for the Electrically Scanning Microwave Radiometer ESMR/Nimbus-6 Image Product

ESMRN6IM

Last Revised 07/12/2022

Goddard Earth Sciences Data and Information Services Center (GES DISC)
http://disc.gsfc.nasa.gov
NASA Goddard Space Flight Center
Code 610.2
Greenbelt, MD 20771 USA

Prepared By:

James E. Johnson	07/12/2022
Name GES DISC GSFC Code 610.2	Date

Reviewed By:

Name	mm/dd/yyyy
Name GSFC Code xxx	Date
Name	mm/dd/yyyy
Name GSFC Code xxx	Date

Goddard Space Flight Center
Greenbelt, Maryland

Revision History

Revision Date	***Changes***	***Author***
07/12/2022	Original	James E. Johnson

Table of Contents

1. Introduction

This document provides basic information on the Nimbus-6 Electrically Scanning Microwave Radiometer (ESMR) Image data. Note, no level 1 radiance data were archived.

1.1 Data Product Description

The ESMR/Nimbus-6 Image product contains brightness temperatures, supplied by the experimenter, displayed as black and white images at either full vertical scale (F) or partial vertical scale (P). In the F mode, up to 125 min of data are output on a single image. In the P mode, data are displayed at twice the vertical scale used at the F mode, and two images are usually needed to display all 125 min of data. Each display contains swaths of horizontally or vertically polarized data of the 37-GHz channel at a brightness temperature range. Spatial resolution at nadir is 20 km in the cross-track direction and 45 km along track. Images are available from June 22, 1975 (day of year 173) through August 11, 1977 (day of year 223).

This product was previously available from the NASA National Space Science Data Center (NSSDC) under the name Electrically Scanning Microwave Radiometer (ESMR) Data, with the identifier ESAD-00201 (old id 75-052A-03B).

1.1.1 The Electrically Scanning Microwave Radiometer

The Nimbus 6 Electrically Scanning Microwave Radiometer (ESMR) measured the earth's microwave emission to provide the liquid water content of clouds, the distribution and variation of sea ice cover, and gross characteristics of land surfaces (vegetation, soil moisture, and snow cover). The two-channel scanning radiometer operated in a 250-MHz band centered at 37 GHz. One channel was used to measure the vertical polarization and the other measured the horizontal polarization. The antenna beam array, a 90- by 20- by 12-cm box-like structure, was mounted on top of the spacecraft sensory ring and was pointed in the direction of the spacecraft's forward motion and tilted down 45 deg from the satellite antenna axis. The antenna beam scanned the earth in 71 discrete steps for various angles extending up to 35 deg on either side of the orbital plane. The deduced brightness temperatures were expected to be accurate to within 3-5 deg K. Spatial resolution was 20 km in the cross-track direction and 45 km in the direction parallel to the subpoint track.

Performance was satisfactory until September 15, 1976, when the horizontal channel output was zero due to a failure of the Ferrite-Dicke switch. The Nimbus-6 ESMR mission was preceded by the ESMR instrument flown aboard Nimbus-5 in 1972.

The original principal investigator for the ESMR experiment was Dr.Thomas T. Wilheit, Jr.

1.1.2 Nimbus-6 Overview

The Nimbus-6 satellite was successfully launched on June 12, 1975. The spacecraft included nine experiments: (1) a Temperature-Humidity Infrared Radiometer (THIR) for measuring day and night surface and cloud top temperatures, as well as the water vapor content of the upper atmosphere, (2) a High-Resolution Infrared Radiation Sounder (HIRS) for determining vertical temperature profiles, and the distribution of water vapor in the atmosphere, (3) the Scanning Microwave Spectrometer (SCAMS) for obtaining vertical profiles of temperature in the troposphere and abundances of liquid water and water vapor, (4) an Electrically Scanning Microwave Radiometer (ESMR) for determining liquid water content of clouds, the distribution and variation of sea ice cover, and land surfaces characteristics, (5) the Earth Radiation Budget (ERB) experiment for accurate measurements of radiation from the sun and earth, (6) a Limb Radiance Inversion Radiometer (LRIR) for determining the vertical distribution of temperature, ozone and water vapor in the stratosphere and lower mesosphere, (7) a Pressure Modulator Radiometer (PMR) for measuring the temperature structure of the upper stratosphere and mesosphere, (8) the Tropical Wind Energy Conversion and Reference Level Experiment (TWERLE) for determining upper atmospheric winds in the tropics, pressure gradients, and provide a reference level in-coordination with in-situ balloon measurements and (9) a Tracking and Data Relay Experiment (T&DRE) for demonstrating data communication from a low-orbiting spacecraft through a synchronous spacecraft to a ground telemetry station.

The orbit of the satellite can be characterized by the following:

- circular orbit at 1100 km
- inclination of 100 degrees
- period of an orbit is about 107.3 minutes
- orbits cross the equator at 26 degrees of longitude separation
- sun-synchronous

1.2 Algorithm Background

The Nimbus-6 ESMR data were generated from the spacecraft telemetry, attitude and orbital data. The data were originally processed on IBM 360 computers. The data were then copied to images and saved on 70 mm film strips. Detailed information on the ESMR data processing can be found in the Nimbus-6 Users' Guide Section 5.

1.3 Data Disclaimer

The data should be used with care and one should first read the Nimbus-6 User's Guide, section 5 describing the ESMR experiment. Users should cite this data product in their research.

Thomas T. Wilheit, Jr. (2022), ESMR/Nimbus-6 Images of Brightness Temperature on 70 mm Film V001, Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC), Accessed: [Data Access Date], 10.5067/LUUX0GAITKXJ

2. Data Organization

The ESMR/Nimbus-6 Image product spans the time period from June 11, 1975 to August 11, 1977.

2.1 File Naming Convention

The data product files are named according to the following convention:

<Platform>_Box<Num>_<BeginOrbit>_<EndOrbit>_<Instrument><Type>_<Sequence>.<Suffix>

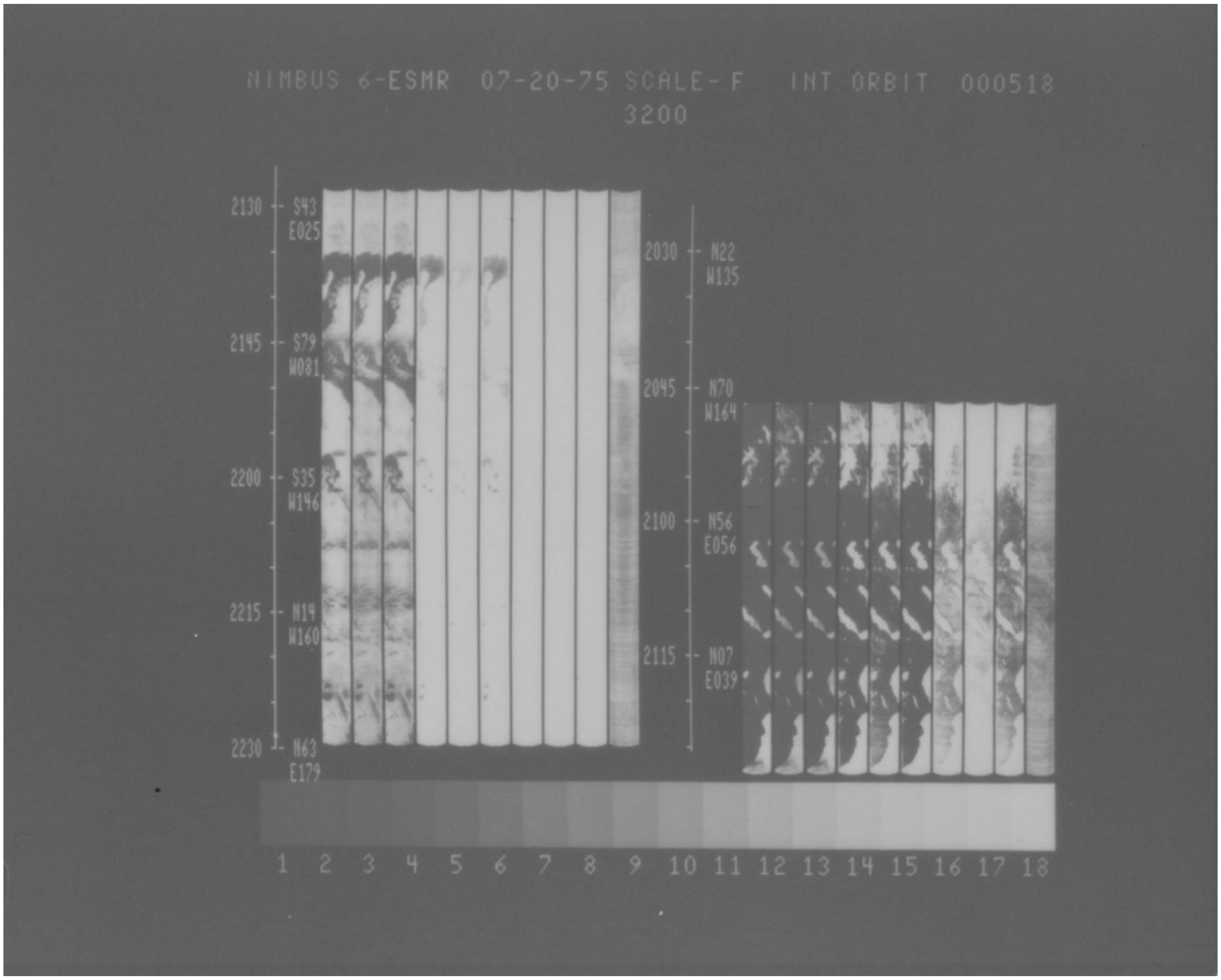
where:

- Platform = name of the platform or satellite (always N6)
- Num = number of box originally containing film roll (18 or 19)
- BeginOrbit = 2-5 digit integer
- EndOrbit = 2-5 digit integer
- Instrument = name of the instrument (always E = ESMR)
- Type = film type (Pos = positive or Neg = Negative)
- Sequence = sequence number of scan in zip directory, 3 digit integer
- Suffix = the file format (always tif, indicating TIFF file)

```
File name example: N6_Box18_140_3287_Epos_011.tif
```

2.2 Image Format and Structure

Image scans are stored as TIFF files and have been combined into eight ZIP files. Four of these contain film negatives, and the other four contain film positives. Each ZIP file contains about 600 to 800 scans, with each scan containing 2 to 3 full images. When unzipping the ZIP files there will be a directory containing two subdirectories: TIF with the scanned images, and NIMBUS_SUPPORT with information about how the images were scanned. Each scan is about 10200 x 2778 pixels, and the 2-3 images within are about 2650 x 2125 pixels in size. The original image prints were 5” x 4” in size. The Nimbus-6 ESMR image layout is shown below:



The image contains the following information on the top line:

- **`NIMBUS 6-ESMR`**: satellite and instrument identifier
- **`DATE`**: 2 digit month, day and year
- **`SCALE`**: F (full 125 minutes of data) or P1/P2 (partial part 1/part2 65 minutes of data)
- **`INT. ORBIT`**: 5 digit interrogation orbit number

In the middle are 10 swath strips on the right (descending node) and 10 swath strips on the left (ascending node) and preceding each a line with hash marks with time (hhmm UTC) at 5 minute increments and sub-satellite latitude/longitude points. There are 71 scan spots across the swath. Each of the 10 swath strips represent:

`H-01, H-04, H-07`: temperature ranges from the horizontal polarization channel
`V-02, V-05, V-08`: temperature ranges from the vertical polarization channel
`S-03, S-06, S-09`: average of the horizontal and vertical polarization temperatures
`D-10`: difference of the horizontal and vertical polarization temperatures

At the bottom is the 18 step gray scale. See Nimbus-6 Users Guide for data value range for each step (different for each swath strip).

2.3 Key Science Data Fields

The primary science data fields in these images are brightness temperatures in Kelvin.

3. Data Contents

The granularity of this data collection is one orbit (approximately 107 minutes).

4. Reading the Data

The image scans can be read using any software package that is able to display TIFF files. Individual TIFF files need to be unzipped using ZIP software.

5. Data Services

5.1 GES DISC Search

The GES DISC provides a keyword, spatial, temporal and advanced (event) searches through its unified search and download interface:

https://disc.gsfc.nasa.gov/

5.2 Documentation

The data product landing page provides information about the data product, as well as links to download the data files and relevant documentation:

https://disc.gsfc.nasa.gov/datacollection/ESMRN6IM_001.html

5.3 Direct Download

The data product is available for users to download directly using HTTPS:

https://acdisc.gesdisc.eosdis.nasa.gov/data/Nimbus6_ESMR_Level1/ESMRN6IM.001/

6. More Information

6.1 Contact Information

Name: GES DISC Help Desk
URL: https://disc.gsfc.nasa.gov/
E-mail: gsfc-help-disc@lists.nasa.gov
Phone: 301-614-5224
Fax: 301-614-5228
Address: Goddard Earth Sciences Data and Information Services Center
Attn: Help Desk
Code 610.2
NASA Goddard Space Flight Center
Greenbelt, MD 20771, USA

6.2 References

T. Wilheit, "The Nimbus-6 User’s Guide - Section 5: The Electrically Scanning Mapping Radiometer (ESMR) Experiment", NASA Goddard Space Flight Center, February 1975,
Pages 87-108

“The Nimbus-6 Data Catalog - Volumes 1-12”, NASA Goddard Space Flight Center, November 1975 to March 1978

7. Appendices

7.1 Acknowledgments

The Nimbus data recovery task at the GES DISC is funded by NASA’s Earth Science Data and Information System program.

7.2 Acronyms

EOS: Earth Observing System

ESDIS: Earth Science and Data Information System

ESMR: Electrically Scanning Mapping Radiometer

GES DISC: Goddard Earth Sciences Data and Information Services Center

GSFC: Goddard Space Flight Center

L1: Level-1 Data

NASA: National Aeronautics and Space Administration

QA: Quality Assessment

TIFF: Tag Image File Format

UT: Universal Time

7.4 FORTRAN Code

```
C-----------------------------------------------------------------------
C ^NAME: READ_SCMR
C
C ^DESCRIPTION:
C     This program reads a Nimbus-5 SCMR level-1 data file and prints
C     contents of the file to the screen. Data files consist of blocks
C     containing records of 8000 bytes, the first is a header record
C     followed by up to 4220 data records. For a description of SCMR
C     see the Nimbus-5 User's Guide, Section 3.
C
C ^MAJOR VARIABLES:
C     FNAME  - name of input file
C     BLOCK  - buffer for data block typically has 50 data records
C     BUFF   - buffer for holding temporary 4-byte word
C     WORD   - integer 4-byte word
C     IBLKSZ - size of block in bytes
C     NRECS  - number of data records per block
C     IOS    - I/O status number
C
C ^NOTES:
C     Compile: gfortran -o READ_SCMR.EXE READ_SCMR.FOR
C
C ^ORGANIZATION: NASA/GSFC, Code 610.2
C
C ^AUTHOR:  James Johnson
C
C ^ADDRESS: james.johnson@nasa.gov
C
C ^CREATED: June 08, 2020
C-----------------------------------------------------------------------

      CHARACTER FNAME*1024                       ! Name of input file
      CHARACTER BLOCK(32000)                     ! Record buffer = 32000 bytes
      CHARACTER BUFF(4)                          ! Buffer to hold 4-byte word
      INTEGER*4 WORD                             ! 4-byte word
      INTEGER*4 IBLKSZ                           ! Block size
      INTEGER*4 IRECSZ /8000/                    ! Record size (8000 bytes)
      INTEGER*4 NRECS                            ! Number of records in block
      EQUIVALENCE (BUFF, WORD)

C     Get the name of the input data file to read
      PRINT *, 'Enter the name of the input file:'
      READ (5,'(A)') FNAME
      PRINT '("FILE: ",A)',TRIM(FNAME)

C     Open the specified input file
      OPEN (UNIT=1, FILE=FNAME, STATUS='OLD', ACCESS='DIRECT',
     &      FORM='UNFORMATTED', RECL=1, ERR=99, IOSTAT=IOS)

C     Initialize M (record block), N (record) and IOFF (byte offset in file)
      M=1
      N=1
      IOFF=0
```

```
C     Loop through the file reading all records in file
      DO

C       Read the first 4-byte word or record size header
        DO I=1,4
          READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) BUFF(I)
        END DO
        IOFF=IOFF+(I-1)
        IBLKSZ = IAND(WORD,'7FFFFFFF'Z)

C       End-of-File (EOF) mark
        IF (IBLKSZ.EQ.0) GOTO 90

C       Bad block size
        IF (MOD(IBLKSZ,8000).NE.0) THEN
          PRINT '("WARN: BAD BLOCK ",I4,", SIZE ",I10)',M,IBLKSZ
        ENDIF

C       Next read the block of records
        DO I=1,IBLKSZ
          READ (1, REC=IOFF+I, IOSTAT=IOS) BLOCK(I)
        END DO
        IOFF=IOFF+(I-1)

        NRECS = IBLKSZ/IRECSZ
        DO J=1,NRECS
          PRINT '("*********************")'
          PRINT '("RECORD     = ",X,I7)', N
          PRINT '("*********************")'
          IF (N.EQ.1) THEN
            CALL PRHREC(BLOCK((J-1)*8000+1:J*8000))
          ELSE
            CALL PRDREC(BLOCK((J-1)*8000+1:J*8000))
          END IF
          N=N+1
        END DO

C       Finally read the last 4-byte word (should match first block size)
        DO I=1,4
          READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) BUFF(I)
        END DO
        IOFF=IOFF+(I-1)

        IF (IBLKSZ.NE.IAND(WORD,'7FFFFFFF'Z)) THEN
          PRINT '("WARN: BLOCK ",I4,", "I11," != ",I11)', M,WORD,IBLKSZ
        ENDIF

        M=M+1
      END DO

C     Close the input file
   90 CLOSE(1)
      STOP

   99 PRINT '("ERROR: OPEN FILE, IOSTAT: ",I6)', IOS
  100 STOP
      END
```

```
C-----------------------------------------------------------------------
C ^SUBROUTINE: PRHREC
C
C     This Subroutine will Print the Header Record
C-----------------------------------------------------------------------
      SUBROUTINE PRHREC(BUFF)

      CHARACTER          BUFF(8000)     ! Record Byte Buffer
      CHARACTER*160      STRBUF         ! String Buffer
      INTEGER*4          IWORD(2000)    ! Array of Words
      REAL*4             RWORD(256)     ! Array of IBM Floats
      INTEGER*4          I4SWAP         ! Function to swap 4-byte words
      REAL*4             R4IBM          ! Function to convert IBM float

      DO I=1,2000
        IWORD(I) = I4SWAP(BUFF((I-1)*4+1:I*4))
      END DO

      DO I=1,160
        STRBUF(I:I) = CHAR(IEBC(ICHAR(BUFF(I))))           ! EBCDIC chars
      END DO
      PRINT '("DATAID     =",X,A160)', STRBUF
      DO I=1,256
        RWORD(I) = R4IBM(IWORD(I+40))
      END DO
      PRINT '("TEMP88     =",/,6(X,G12.6))', RWORD         ! WORDS   41-296
      DO I=1,256
        RWORD(I) = R4IBM(IWORD(I+296))
      END DO
      PRINT '("RAD88      =",/,6(X,G12.6))', RWORD         ! WORDS  297-552
      DO I=1,256
        RWORD(I) = R4IBM(IWORD(I+552))
      END DO
      PRINT '("TEMP109    =",/,6(X,G12.6))', RWORD         ! WORDS  553-808
      DO I=1,256
        RWORD(I) = R4IBM(IWORD(I+808))
      END DO
      PRINT '("RAD109     =",/,6(X,G12.6))', RWORD         ! WORDS  809:1064
      DO I=1,256
        RWORD(I) = R4IBM(IWORD(I+1064))
      END DO
      PRINT '("VOLT12     =",/,6(X,G12.6))', RWORD         ! WORDS 1065-1320
      DO I=1,256
        RWORD(I) = R4IBM(IWORD(I+1320))
      END DO
      PRINT '("RAD12      =",/,6(X,G12.6))', RWORD         ! WORDS 1321-1576
      DO I=1,8
        STRBUF(I:I) = CHAR(IEBC(ICHAR(BUFF(I+6304))))
      END DO
      PRINT '("CALPRODATE =",X,A8)', STRBUF
      DO I=1,12
        STRBUF(I:I) = CHAR(IEBC(ICHAR(BUFF(I+6312))))
      END DO
      PRINT '("CALPROTIME =",X,A12)', STRBUF
      PRINT '("NSAMPLEDEG =",X,F8.2)', R4IBM(IWORD(1783))
      PRINT '("SAMPLE0DEG =",X,F8.2)', R4IBM(IWORD(1784))
      PRINT '("UNKNOWN    =",/,6(X,G12.6))',
     +                                     (R4IBM(IWORD(I)),I=1785,1834)
c     PRINT '("SPARES     =",16(X,I3))', (ICHAR(BUFF(I)),I=7337,8000)

      RETURN
      END

C-----------------------------------------------------------------------
C ^SUBROUTINE: PRDREC
C
C     This Subroutine will Print the Data Records
C-----------------------------------------------------------------------

      SUBROUTINE PRDREC(BUFF)

      CHARACTER          BUFF(8000)     ! Record Byte Buffer
      INTEGER*4          IWORD(2000)    ! Array of Words
      INTEGER*4          I4SWAP         ! Function to swap 4-byte words
      REAL*4             R4IBM          ! Function to convert IBM float

      DO I=1,2000
        IWORD(I) = I4SWAP(BUFF((I-1)*4+1:I*4))
      END DO

      PRINT '("DAY        =",X,I11)', IWORD(1)
      PRINT '("MILLISEC   =",X,I11)', IWORD(2)
      PRINT '("CHANNELID  =",X,I11)', ISHFT(IWORD(3),-4)
      PRINT '("DATAFLAG   =",X,I11)', IAND(IWORD(3),'0F'Z)
      PRINT '("DATA       =",/,20(X,I3))', (ICHAR(BUFF(I)),I=13,6960)
      PRINT '("GHA        =",X,G12.6)', R4IBM(IWORD(1741))
      PRINT '("SCLAT      =",X,G12.6)', R4IBM(IWORD(1742))
      PRINT '("SCLON      =",X,G12.6)', R4IBM(IWORD(1743))
C     PRINT '("SPARE      =",4(X,I3))', (ICHAR(BUFF(I)),I=6973,6976)
      PRINT '("SCHEIGHT   =",X,G12.6)', R4IBM(IWORD(1745))
      PRINT '("DAYNIGHT   =",X,I12)', IWORD(1746)
C     PRINT '("SPARES     =",16(X,I3))', (ICHAR(BUFF(I)),I=6985,7000)
      PRINT '("LAT        =",/,6(X,G12.6))',
     +                                     (R4IBM(IWORD(I)),I=1751,1851)
      PRINT '("LON        =",/,6(X,G12.6))',
     +                                     (R4IBM(IWORD(I)),I=1852,1952)
C     PRINT '("SPARES     =",/,20(X,I3))', (ICHAR(BUFF(I)),I=7809,8000)

      RETURN
      END

C-----------------------------------------------------------------------
C ^FUNCTION: I4SWAP
C
C     This function will swap the bytes of a 4-byte word
C-----------------------------------------------------------------------

      INTEGER*4 FUNCTION I4SWAP(BUFF)

      CHARACTER          BUFF(4)        ! Input data buffer
      CHARACTER          TEMP(4)        ! Output swapped buffer
      INTEGER*4          I4BUFF
      EQUIVALENCE        (TEMP, I4BUFF)
      TEMP(1) = BUFF(4)
      TEMP(2) = BUFF(3)
      TEMP(3) = BUFF(2)
      TEMP(4) = BUFF(1)
      I4SWAP = I4BUFF

      RETURN
      END

C-----------------------------------------------------------------------
C ^FUNCTION: R4IBM
C
C     This function will convert an input word to an IBM float
C-----------------------------------------------------------------------

      FUNCTION R4IBM(IWORD)

      INTEGER*4      IWORD              ! input word
      INTEGER*4      IDROW              ! reverse the bits of input word
      REAL*8         A /16.0/           ! base number
      INTEGER*4      B /64/             ! exponent offset
      REAL*8         C /0.0/            ! fraction offset
      INTEGER*1      S                  ! sign flag
      INTEGER*2      E                  ! binary exponent
      REAL*8         F                  ! binary fraction
      REAL*8         M                  ! mantissa
      REAL*8         V                  ! float value
      INTEGER*4      I                  ! counter

      S = ISHFT(IWORD, -31)

      E = 0
      DO 10 I=0,6
        E = E + IAND(ISHFT(IWORD,-24),ISHFT(1,I))
   10 END DO

      IDROW = 0
      DO 11 I=0,31
        IF (IAND(IWORD,ISHFT(1,I)) .NE. 0) THEN
          IDROW = IOR(IDROW,(ISHFT(1,31-I)))
        END IF
   11 END DO

      F = 0.0
      DO 12 I=0,31
        IF (ISHFT(IAND(ISHFT(IDROW,-8),ISHFT(1,I)),1) .NE. 0) THEN
          F = F + 1 / FLOAT(ISHFT(IAND(ISHFT(IDROW,-8),ISHFT(1,I)),1))
        END IF
   12 END DO

      M = C + F                         ! calculate the mantissa
      V = (-1)**S * M * A**(E - B)      ! calculate the float value
      IF (ABS(V) .LT. 2.0**(-149)) THEN
        V = (-1)**S * 0.0               ! avoid underflow
      END IF
      R4IBM = V
      RETURN
      END

C-----------------------------------------------------------------------
C     This Function returns EBCDIC to ASCII character index
C     (Non-ASCII characters use '1A'x (CTRL+Z) = substitute character)
C-----------------------------------------------------------------------
      FUNCTION IEBC(I)

      INTEGER   EBCTBL(256)

      DATA EBCTBL /
     + Z'00',Z'01',Z'02',Z'03',Z'1A',Z'09',Z'1A',Z'7F',    ! 00-07
     + Z'1A',Z'1A',Z'1A',Z'0B',Z'0C',Z'0D',Z'0E',Z'0F',    ! 08-0F
     + Z'10',Z'11',Z'12',Z'13',Z'1A',Z'1A',Z'08',Z'1A',    ! 10-17
     + Z'18',Z'19',Z'1A',Z'1A',Z'1C',Z'1D',Z'1E',Z'1F',    ! 18-1F
     + Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'0A',Z'17',Z'1B',    ! 20-27
     + Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'05',Z'06',Z'07',    ! 28-2F
     + Z'1A',Z'1A',Z'16',Z'1A',Z'1A',Z'1A',Z'1A',Z'04',    ! 30-37
     + Z'1A',Z'1A',Z'1A',Z'1A',Z'14',Z'15',Z'1A',Z'1A',    ! 38-3F
     + Z'20',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! 40-47
     + Z'1A',Z'1A',Z'5B',Z'2E',Z'3C',Z'28',Z'2B',Z'21',    ! 48-4F
     + Z'26',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! 50-57
     + Z'1A',Z'1A',Z'5D',Z'24',Z'2A',Z'29',Z'3B',Z'5E',    ! 58-5F
     + Z'2D',Z'2F',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! 60-67
     + Z'1A',Z'1A',Z'7C',Z'2C',Z'25',Z'5F',Z'3E',Z'3F',    ! 68-6F
     + Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! 70-77
     + Z'1A',Z'60',Z'3A',Z'23',Z'40',Z'27',Z'3D',Z'22',    ! 78-7F
     + Z'1A',Z'61',Z'62',Z'63',Z'64',Z'65',Z'66',Z'67',    ! 80-87
     + Z'68',Z'69',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! 88-8F
     + Z'1A',Z'6A',Z'6B',Z'6C',Z'6D',Z'6E',Z'6F',Z'70',    ! 90-97
     + Z'71',Z'72',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! 98-9F
     + Z'1A',Z'7E',Z'73',Z'74',Z'75',Z'76',Z'77',Z'78',    ! A0-A7
     + Z'79',Z'7A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! A8-AF
     + Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! B0-B7
     + Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! B8-BF
     + Z'7B',Z'41',Z'42',Z'43',Z'44',Z'45',Z'46',Z'47',    ! C0-C7
     + Z'48',Z'49',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! C8-CF
     + Z'7D',Z'4A',Z'4B',Z'4C',Z'4D',Z'4E',Z'4F',Z'50',    ! D0-D7
     + Z'51',Z'52',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! D8-DF
     + Z'5C',Z'1A',Z'53',Z'54',Z'55',Z'56',Z'57',Z'58',    ! E0-E7
     + Z'59',Z'5A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',    ! E8-EF
     + Z'30',Z'31',Z'32',Z'33',Z'34',Z'35',Z'36',Z'37',    ! F0-F7
     + Z'38',Z'39',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A',Z'1A'/    ! F8-FF

      IEBC = EBCTBL(I+1)

      RETURN
      END
```