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

README Document for the Surface Composition Mapping Radiometer SCMR/Nimbus-5 Level-1 Calibrated Brightness Temperature Product

SCMRN5L1

Last Revised 03/29/2022

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	03/29/2022
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
03/29/2022	Original	James E. Johnson

Table of Contents

1. Introduction

This document provides basic information on using the Surface Composition Mapping Radiometer SCMR/Nimbus-5 Level-1 Earth Surface Radiance data.

1.1 Data Product Description

The SCMR/Nimbus-5 Level-1 Earth Surface Radiance product contains calibrated and located IR radiances and brightness temperatures at 660 x 660 m ground resolution. Data are grouped in 7-min observations in the Mercator projection, covering globally from 80 deg S. to 80 deg N. Files also contain, quality flags, geolocation, orbital, and housekeeping information. Each file typically contains a 7 minute scene of data. The data are available from December 11, 1972 through December 30, 1972.

This product was previously available from the NASA National Space Science Data Center (NSSDC) under the name Surface Composition Mapping Radiometer (SCMR) Data, with the identifier ESAD-00128 (old id 70-025A-05B).

1.1.1 The Surface Composition Mapping Radiometer

The Surface Composition Mapping Radiometer (SCMR) measured (1) terrestrial radiation in the 8.3- to 9.3-micrometer and 10.2- to 11.2-micrometer intervals and (2) reflected solar radiation in the 0.8- to 1.1-micrometer range. Surface composition and sea surface temperatures could be obtained from these measurements. The SCMR had an instantaneous field of view (FOV) of 0.6 mrad, equivalent to a ground resolution of 660 m at nadir. The scan mirror rotated at 10 rps to provide scan lines 800-km wide across the spacecraft track.

The last usable data were transmitted on January 4, 1973 (orbit 320). A modified instrument, the Heat Capacity Mapping Radiometer, was flown on the Heat Capacity Mapping Mission (HCMM) in 1978.

The original principal investigator for the SCMR experiment was Dr. Warren A. Hovis from NOAA NESDIS.

1.1.2 Nimbus-5 Overview

The Nimbus 5 satellite was successfully launched on December 11, 1972. The primary experiments included: (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) an Electrically Scanning Microwave Radiometer (ESMR) for mapping the microwave radiation from the Earth's surface and atmosphere, (3) an Infrared Temperature Profile Radiometer (ITPR) for obtaining vertical profiles of temperature and moisture, (4) the Nimbus-E Microwave Spectrometer (NEMS) for determining tropospheric temperature profiles, atmospheric water vapor abundances, and cloud liquid water contents, (5) a Selective Chopper Radiometer (SCR) for observing the global temperature structure of the atmosphere, and (6) a Surface Composition Mapping Radiometer (SCMR) for measuring the differences in the thermal emission characteristics of the Earth's surface.

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

- circular orbit at 1100 km
- inclination of 99.9 degrees
- period of an orbit is about 107.2 minutes
- orbits cross the equator at 26 degrees of longitude separation
- sun-synchronous

1.2 Algorithm Background

The Nimbus-5 SCMR data were generated from the spacecraft telemetry, attitude and orbital data. The data were originally processed on IBM 360 computers. The intensity of infrared terrestrial radiation centered at 8.8 microns and 10.9 microns, and reflected solar radiation at 1.2 microns are stored along with geolocation, ancillary and other housekeeping data. Detailed information on the SCMR data processing can be found in the Nimbus-5 Users' Guide Section 3 and the SCMR Data Processing System documents.

1.3 Data Disclaimer

The data should be used with care and one should first read the Nimbus-5 User's Guide, section 3 describing the SCMR experiment, as well as the SCMR Data Processing System documents. Users should cite this data product in their research.

Hovis, Warren A. (2022), SCMR/Nimbus-5 Level 1 Calibrated and Geolocated Radiances V001, Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC), Accessed: [Data Access Date], https://doi.org/10.5067/252PDBEURQZE

1.4 Known Issues

During review of the data, the information stored in the header record at bytes 7137 – 7336 could not be determined from the available documentation. It was determined that the values are 50 Real*4 values. The first 160 bytes of the header record are specified as being a data identification from the raw data tape in EBCDIC characters, though it doesn't seem to be true as several values are not valid EBCDIC characters. Also, while plotting the data it appears that the latitude and longitude values are slightly offset by up to perhaps 1/2 degree depending on the position in the data scan.

2. Data Organization

The SCMR/Nimbus-5 Level-1 Calibrated Radiance Tape product spans the time period from December 11, 1972 to December 30, 1972. Each file typically contains up to 7 minutes of data.

2.1 File Naming Convention

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

<Platform>-<Instrument>_<Level>_<Date>_<Tape>.<Suffix>

where:
- Instrument = name of the instrument (always SCMR)
- Platform = name of the platform or satellite (always Nimbus4)
- Level = processing level of data (always L1)
- Date = Data start date and time in UTC in format <YYYY>m<MMDD>t<hhmm> where
    1. YYYY = 4 digit year (1972)
    2. MM = 2 digit month (01-12)
    3. DD = 2 digit day of month (01-31)
    4. hh = 2 digit hour of day (01-23)
    5. mm = 2 digit minute of hour (01-59)
- Tape = tape number (DR primary tape, DS backup tape plus 4 digit number)
- Suffix = the file format (always TAP, indicating binary data tape file)

```
File name example: Nimbus5-SCMR_L1_1972m1220t020005_DS3684.TAP
```

2.2 File Format and Structure

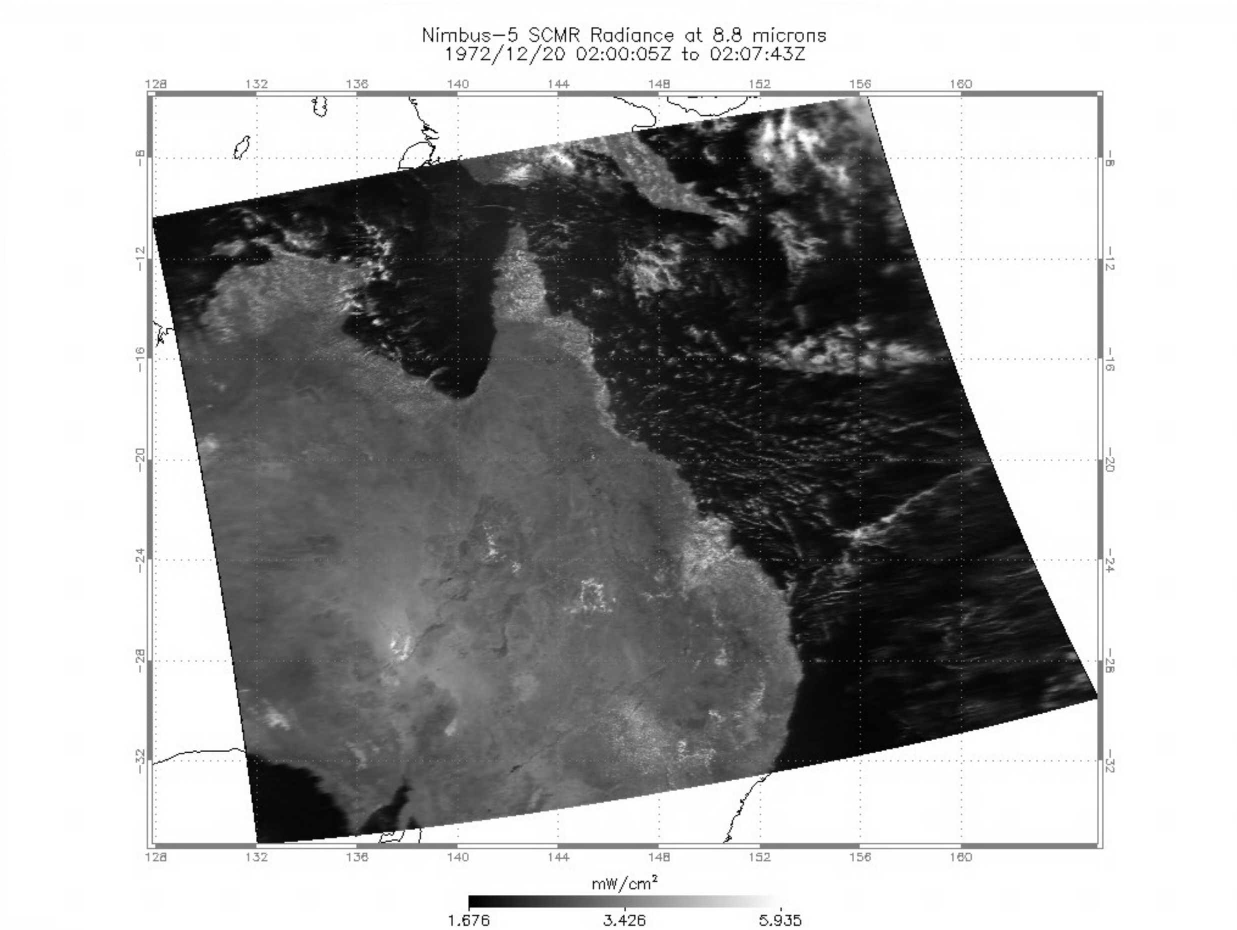
The data are stored as they were originally written in IBM binary (big-endian) record oriented structured files. Each original tape held up to seven data files containing up to 7 minutes worth of measurements. A tape consists of multiple data files. Each file has many record blocks, each preceeded by an initial marker (4-byte integer) with the size of the block, and ended with another marker, which should match the initial one. A data file is ended with a single end-of-file marker (4 byte integer) with size zero. The last file on the tape will end with two end-of-file marker, i.e. end-of-tape marker. The files were originally written on 1600 bpi 9-track tapes using a FORTRAN block format, each record block will have a multiple of 8000 byte records (up to 4 records or a maximum of 32000 byte blocks).

This data collection consisted of data from 3 primary tapes (designated with a DR and four digits) and 8 backup tapes (designated with DS and four digits). There were 30 files on the DR tapes, and 39 files on the DS tapes. Of these, several files were found to be duplicates such that in the end just 37 data files total were unique, 27 from the primary DR tapes and 10 from the backup DS tapes.

2.3 Key Science Data Fields

The primary science data fields in this data product are the calibrated infrared radiances in units of W/cm^2 and temperatures in units of degrees Kelvin.

Figure 2: Typical data coverage of a Nimbus-5 SCMR Level-1 data file.



3. Data Contents

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

3.1 Data Records

The Level-1 data files contain a header record followed by a series of up to 4200 data records. Each record is of size 8000 bytes (2000 x 4-byte integer words), a record block will contain a maximum of 4 logical records. See Appendix C in the Nimbus-5 SCMR Data Processing Document for a description of the SCMR calibrated file format (note it has errors in byte/word offsets, which were corrected here as best as possible). A sample FORTRAN program to read the data files can be found in the Appendix of this README.

Table 3-1-1: Header Record

Bytes	Word	Format	Description	
1 – 160	1 – 40	160 x EBCDIC	Data Identification from Raw Data	
161 – 1184	41 – 296	256 x R*4	8.8 µm Temperature	Master Conversion Tables
1185 – 2208	297 – 552	256 x R*4	8.8 µm Radiance	
2209 – 3232	553 – 808	256 x R*4	10.9 µm Temperature	
3233 – 4256	809 – 1064	256 x R*4	10.9 µm Radiance	
4257 – 5280	1065 – 1320	256 x R*4	1.2 µm Voltage	
5281 – 6304	1321 – 1576	256 x R*4	1.2 µm Radiance	
6305 – 6312	1577 – 1578	8 x EBCDIC	Calibration Processing Date (mm/dd/yy)	
6313 – 6324	1579 – 1581	12 x EBCDIC	Calibration Processing Time (HH:MM:SS.sss)	
6325 – 7128	1582 – 1782	804 bytes	Unused (set to zero)	
7129 – 7132	1783	R*4	Number of Samples per Degree Nadir Angle	
7133 – 7136	1784	R*4	Sample Corresponding to 0 Degree Nadir Angle	
7137 – 7336	1785 – 1834	50 x R*4	Unknown (related to Nadir Angles?)	
7337 – 8000	1835 – 2000	664 bytes	Unused (set to zero)	

Table 3-1-2: Data Record

Bytes	Word	Format	Description
1 – 4	1	I*4	Day of Year
5 – 8	2	I*4	Time (milliseconds)
9 – 10	3A	I*2	Channel Indicator (0 = 8.8µm; 1=1.2µm)
11 – 12	3B	I*2	Data Flag from Raw Data Record
13 – 6960	4 – 1740	6948 bytes	Index to Master Conversion Table (see header) Interleaved: odd (13, 15, etc.) = 8.8µm or1.2µm (see channel indicator), even (14, 16, etc.) = 12µm
6961 – 6964	1741	R*4	Greenwich Hour Angle (degrees)
6965 – 6968	1742	R*4	Sub-Satellite Point Latitude (degrees + 90°)
6969 – 6972	1743	R*4	Sub-Satellite Point Longitude (degrees West)
6973 – 6976	1744		Unused (zero filled)
6977 – 6980	1745	R*4	Spacecraft Height (km)
6981 – 6984	1746	R*4	Day/Night/Twilight Indicator: 0=day, 1=twilight, 2=night
6985 – 7000	1747 - 1750		Unused (zero filled)
7001 – 7404	1751 – 1851	101 x R*4	Latitude (degrees + 90°) of Each Nadir Angle
7405 – 7808	1852 – 1952	101 x R*4	Longitude (degrees West) of Each Nadir Angle
7809 – 8000	1953 – 2000		Unused (zero filled)

3.2 Metadata

The metadata are contained in a separate XML formatted file having the same name as the data file with .xml appended to it.

Table 3-2: Metadata attributes associated with the data file.

Name	Description
LongName	Long name of the data product.
ShortName	Short name of the data product.
VersionID	Product or collection version.
GranuleID	Granule identifier, i.e. the name of the file.
Format	File format of the data file.
CheckSumType	Type of checksum used.
CheckSumValue	The value of the calculated checksum.
SizeBytesDataGranule	Size of the file or granule in bytes.
InsertDateTime	Date and time when the granule was inserted into the archive. The format for date is YYYY-MM-DD and time is hh-mm-ss.
ProductionDateTime	Date and time the file was produced in format YYYY-MM-DDThh:mm:ss.ssssssZ
RangeBeginningDate	Begin date when the data was collected in YYYY-MM-DD format.
RangeBeginningTime	Begin time of the date when the data was collected in hh-mm-ss format.
RangeEndingDate	End date when the data was collected in YYYY-MM-DD format.
RangeEndingTime	End time of the date when the data was collected in hh-mm-ss format.
PlatformShortName	Short name or acronym of the platform or satellite
InstrumentShortName	Short name or acronym of the instrument
SensorShortName	Short name or acronym of the sensor
GPolygon: PointLatitude	Latitudes of the polygon (rectangle) points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
GPolygon: PointLongitude	Longitudes of the polygon (rectangle) points that represent the satellite coverage. Each point is identified by its latitude and longitude pair.
Elapsed_Min_Time	Duration in minutes of data collected during an orbit

4. Reading the Data

The data are written in a binary record-oriented format. Using the record format specification in the section above, users can write software to read the data files. Please note that the data were originally written using a big-endian format, therefore users on little-endian machines will need to swap bytes for the words. Also, the floating point data were written using IBM 360 machines, and must be converted if reading on a machine that understands IEEE floats (integers are not affected).

A sample FORTRAN program is included in the Appendix section which will read the data files.

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/SCMRN5L1_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/Nimbus5_SCMR_Level1/SCMRN5L1.001/

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

W. A. Hovis Jr., "The Nimbus-5 User's Guide - Section 3: The Surface Composition Mapping Radiometer (SCMR) Experiment", NASA Goddard Space Flight Center, November 1972, Pages 49-58

S. K. Tak, "Surface Composition Mapping Radiometer (SCMR) Data Processing System", NASA GSFC, 1973

K. Hanlin, R. Reeder, L. Rhodes, W. Stallings, and S. Willis, "Nimbus-E SCMR Data Processing System," NASA GSFC, October 1972

7. Appendices

▪

7.1 Acknowledgements

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

GES DISC: Goddard Earth Sciences Data and Information Services Center

GSFC: Goddard Space Flight Center

SCMR: Surface Composition Mapping Radiometer

CBTT: Calibrated Brightness Temperature Tape Product

L1: Level-1 Data

NASA: National Aeronautics and Space Administration

EDSC: Earthdata Search Client

QA: Quality Assessment

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
```