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

README Document for the Explorer 7 Thermal Radiation Experiment Level 1 Data Products

EXP7L1TRTALL
EXP7L1TRTWHT

Last Revised 07/30/2021

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

Prepared By:

James E. Johnson	07/30/2021
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*
07/30/2021	Original	James E. Johnson

Table of Contents

1. Introduction

This document provides basic information on using the Explorer 7 Thermal Radiation Experiment Level 1 data products.

1.1 Data Product Description

There are two Explorer 7 Thermal Radiation Experiment Level 1 Data Products. The first product (EXP7L1TRTALL) contains temperature values measured by all sensors, and the second product (EXP7L1TRTWHT) contains temperature values from just the white sensor at nighttime. The data were originally created on IBM 7090 computers and eventually copied to 6250 bpi 7-track magnetic tapes (originally stored on 200 bpi 7-track tapes) . The data from these tapes were recovered from the magnetic tapes and are now archived in digital files in their original binary file formats.

The EXP7L1TRTALL data are available for the time period from 12 July 1961 to 30 September 1961. There are two data files which are in a record oriented binary file format containing the measurements from all sensors. Note there is no geolocation information included with these data.

The EXP7L1TRTWHT data are available for the time period from 19 October 1959 to 4 June 1960. There is just a single file for the entire mission which is written in the IBM Extended BCD text format.

The principal investigator for the Explorer 7 Thermal Radiation Experiment was Verner E. Suomi from the University of Wisconsin. These products were previously available from the NASA National Space Science Data Center (NSSDC) under the names Temperature Values from all Sensors with the identifier ESAD-00249 (old id 59-009A-01B) and Selected White Sensor Temperature (Nighttime) Values with the identifier ESAD-00248 (old id 59-009A-01A).

1.1.1 Thermal Radiation Experiment

The Explorer 7 thermal radiation experiment was designed to measure incident and reflected solar UV radiation and terrestrial IR radiation in order to obtain a better understanding of the driving forces of the earth-atmosphere system. It was the first experiment to observe large scale weather patterns from space.

The primary instrumentation consisted of five bolometers in the form of hollow silver hemispheres that were thermally insulated from, but in close proximity to specially aluminized mirrors. The hemispheres thereby behaved very much like isolated spheres in space. Two of the hemispheres had black coatings and responded about equally to solar and terrestrial radiation. A third hemisphere, coated white, was more sensitive to terrestrial radiation than to solar radiation. A fourth, which had a gold metal surface, was more sensitive to solar radiation than to terrestrial radiation. The fifth Tabor surfaced hemisphere, protected from direct sunlight, was used to measure the reflected

sunlight. A glass-coated bead thermistor was mounted on the top of each hemisphere to measure the temperature.

On the night side of the Earth, useful signals were received for 15 to 20 minutes, and these were signals were more clear than those collected on the day side. The radius of coverage was , increasing to 31.5° (~3500 km) at apogee. Half the radiation power was received from an area below the satellite with a radius of 5.3° (545 km) at perigee and 9° (~1015 km) at apogee. A complete set of four temperature observations and one reference sample required 30 seconds. Thus, in each orbit, about 180 temperature measurements could be obtained.

The Explorer 7 thermal radiation experiment made about 4000 measurements each day and of these between 400 and 1000 were analyzed.

1.1.2 Explorer 7 Overview

Explorer 7 was launched on a Juno II rocket at 15:31 UTC on 13 October 1959 from the Atlantic Missile Range. The spin-stabilized satellite's external structure consisted of two truncated conical fiberglass shells joined by a cylindrical aluminum center section. The spacecraft was 76 cm wide at its equator and about 76 cm high with a payload mass of about 41.5 kg. The spacecraft was powered by approximately 3000 solar cells mounted on both the upper and lower shells. Explorer 7 was designed to measure solar X-ray and Lyman-alpha flux, trapped energetic particles, and heavy primary cosmic rays. Secondary objectives included collecting data on micrometeoroid penetration and molecular sputtering, and studying the Earth-atmosphere heat balance.

There were 6 experiments aboard the Explorer 7 spacecraft:

1) Ground Based Ionospheric
2) Heavy Primary Cosmic Rays
3) Micrometeorite
4) Solar X-Ray and Lyman-Alpha Radiation
5) Thermal Radiation
6) Trapped Radiation and Solar Protons

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

- Perigee Altitude: 573 km
- Apogee Altitude: 1073 km
- Orbital Period: 101.38 minutes
- Inclination: 50.72 degrees
- Eccentricity: 0.03469

1.2 Algorithm Background

The Explorer 7 Thermal Radiation Experiment data were generated from the spacecraft telemetry, attitude and orbital data. The data were originally processed on IBM 7094 computers, and subsequently copied to 200 bpi 7-track tapes for archival. More detailed information on the Explorer 7 Thermal Radation Experiment and data processing can be found in the Explorer 7 (1959 IOTA 1) Thermal Radiation Experiment Data Users' Note.

1.3 Data Disclaimer

The data should be used with care and one should first read the Explorer 7 Thermal Radiation Experiment Data User's Note describing the thermal radiation experiment and data. Users should cite these data products in their research:

Suomi, Verner. (2021), Explorer-7 Thermal Radiation Experiment Temperature Values from All Sensors V001, Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC), Accessed: **[*Data Access Date*]**, https://doi.org/10.5067/NGSARJ2J4XZQ

Suomi, Verner. (2021), Explorer-7 Thermal Radiation Experiment Selected White Sensor Temperature (Nighttime) Values V001, Greenbelt, MD, USA, Goddard Earth Sciences Data and Information Services Center (GES DISC), Accessed: **[*Data Access Date*]**, https://doi.org/10.5067/Y1XFERKNBTHB

2. Data Organization

The Explorer 7 Level 1 data span the time period from October 19, 1959 to June 4, 1960. The white sensor night data file contains data for the entire mission, the all sensors data is split across two files.

2.1 File Naming Convention

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

<Platform>-<Experiment>_<Level>-<Type>_<StartDate>_<EndDate>_<TapeNumber>-<FileNumber>.<Suffix>, where

o) Platform = name of the platform or satellite (Expl7 for Explorer 7)
o) Experiment = experiment name (ThermRad for Thermal Radiation Experiment)
o) Level = process level (L1)
o) Type =- Data type of data (TempAllSensors or TempWhiteSensorNight)
o) StartDate = Data start date in format <YYYY>m<MMDD> where
1. YYYY = 4 digit year (1959 or 1960)
2. MM = 2 digit month (01 - 12)
3. DD = 2 digit day of month (01 – 31)

o) EndDate = Data end date in format <YYYY>m<MMDD> where
4. YYYY = 4 digit year (1960)
5. MM = 2 digit month (01 - 12)
6. DD = 2 digit day of month (01 - 31)

o TapeNumber = 4 digit number of tape (preceded by ‘DR’ - primary or ‘DS’ - backup)
o FileNumber = 3 digit number of file on tape
o Suffix = the file format (always TAP, indicating tape binary data)

```
File names: Expl7-ThermRad_L1-TempAllSensors_1959m1019_1960m0416_DR2858-001.TAP
            Expl7-ThermRad_L1-TempAllSensors_1960m0416_1960m0604_DR2858-002.TAP
            Expl7-ThermRad_L1-TempWhiteSensorNight_1959m1115_1960m0524_DR2834-001.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. The files were last copied to 6250 bpi 9-track tapes using a blocked FORTRAN format. Each tape has one (white sensor nighttime product) or two files (all sensors product) on it. Each data file on the tape contains a set of data records with a FORTRAN record size word, the record block, and a FORTRAN record trailing size word.

The white sensor nighttime product file contains all data for the entire mission. Each record represents one line of IBM extendend BCD text characters (or EBCDIC). The first 17 lines (or records) on the file contain no useful information and can be ignored. Records (lines) of size 84 bytes contain the data. The file ends with an end-of-tape (two words with value 0).

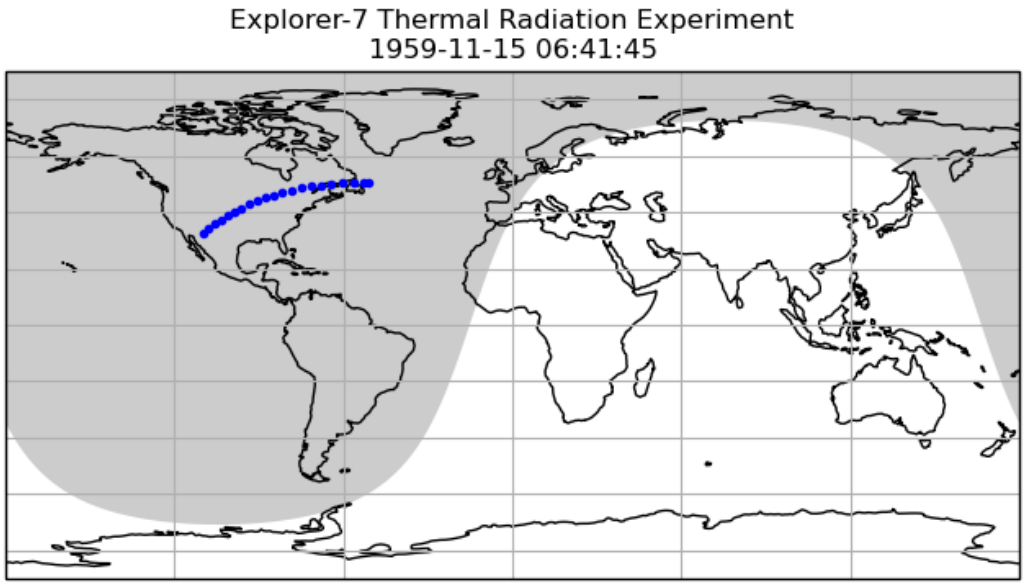
The all sensor product files are written in big-endian IBM 36-bit integers, stored in 4 ½ 8 bytes. The first record on each file is 2764 bytes, contains no useful information and can be skipped. The rest of the data records are 1238 bytes. For the contents and layout of the data, see section 3.1 below.

During data recovery there were two sets of tapes, The first set of tapes are the primary tapes and designated with a DR with a total of 2 tapes containing 3 files. The second set are the backup tapes which are designated with a DS also with a total of 2 tapes and 3 files. During recovery all of the DR and DS tape files were exact duplicates of each other. In the end there were 3 unique files (1 file for white sensor nighttime product and 2 files for all sensors product). Just the primary DR tapes are needed to represent the complete record of the Explorer Level 1 Thermal Radiation Experiment data collections and are publicly available from the GES DISC.

2.3 Key Science Data Fields

The primary science data fields are temperatures expressed in degrees Kelvin.

Figure 1: Typical data coverage for a Explorer 7 white sensor temperature (nighttime) orbit. The all sensor temperature data would be similar but also cover the daytime portion of an orbit (shaded region is nighttime).



3. Data Contents

The granularity for the Explorer 7 data is the entire mission.

3.1 Data Records

The file formats of these data are described in the document “Explorer 7 (IOTA 1) Thermal Radiation Experiment Data User’s Note". A summary of this is described in the two tables below.

The white sensor (nighttime) temperature product contains data records with one line of text in IBM EBCDIC format. The first 17 records of the file can be ignored, those with size 84 bytes contain the data. The first line begins with a character with value 1 (page restore) or 2 (half page restore) followed by a header listing the satellite, experiment and number of days since launch. The next three lines begin with a 0 character (double space) and additional header information: record number and station code, then date and time of the measurement, followed by labels for each column of data to follow. The first character of the lines with data is a blank character (single space), organized as listed in Table 3-1. Headers and data repeat for subsequent nighttime orbit segments (1370 total). There is just one file for the entire mission (November 15, 1959 to May 24, 1960).

Table 3-1: Explorer 7 White Sensor (Nighttime) Temperature Data Record

Column	Symbol	Meaning
1	ISPACE	Carriage Control (blank space)
2-7	E-LONG	Longitude in deg east (decimal column 6)
8-15	N-LAT	Latitude in deg north (decimal column 14)
16-22	RAD	Longwave radiation in Langleys/min (decimal column 18)
23-28	BETA	Solid angle to earth in steradians (decimal column 26)
29-37	SENSOR	Temperature of white sensor in Kelvin (decimal column 35)
38-46	MIRROR	Temperature of mirror in Kelvin; averaged value from REFS column (decimal in column 44)
47-55	REFS	Low frequency reference in Kelvin, 65.0 ± 2.0 High frequency reference in Kelvin, 985.0 ± 15.0 AMBA 1 (skin temperature) in Kelvin AMBA 2 (solar cell temperature) in Kelvin AMBA 3 (transmitter temperature) in Kelvin AMBA 4 (battery temperature) AMBA 5 (Geiger tube temperature) Mirror 1 (for black hemisphere 1 and white) Mirror 2 (for black hemisphere 4 and gold (decimal in column 53)
56-84	BLANK	All blank, though some have a comment, last 4 characters are zeros

Output for a typical orbit segment:

```
EXPLORER VII RADIATION DATA    33 DAYS AFTER LAUNCH

LIBRARY RECORD   233   4   STATION  9

DATE 15/11/1959   TIME  3 25 40 GMT

E-LONG   N-LAT   RAD   BETA   SENSOR   MIRROR    REFS
 -26.4    47.9  .3067  3.38   196.67    274.9   289.73
 -23.0    48.6  .3173  3.41   196.10    274.4   274.25
 -19.5    49.2  .3078  3.42   195.86    274.0   273.67
 -15.9    49.7  .3171  3.44   195.49    273.6    64.00
 -12.2    50.1  .3208  3.45   195.38    273.1   991.00
  -8.4    50.3  .3133  3.48   195.38    272.7   261.50
  -4.6    50.4  .2997  3.50   195.26    272.3   257.85
   -.8    50.3  .3140  3.51   194.90    271.9   290.90
   2.9    50.2  .3015  3.53   194.90    271.4   286.01
   6.7    49.8  .2985  3.54   194.66    271.0   286.86
  10.4    49.4  .3014  3.57   194.42    270.6   270.59
  12.2    49.1  .0000  3.58   194.30    270.1   269.63
```

Note: you have to locate the low frequency and high frequency references, they do not always start with those but the remaining order is the same.

The following records have problems:

3940 – data record line is missing SENSOR, MIRROR and REFS are values

7037 – longitude shifted one column to right, remove extra level of precision (2nd 0 after decimal)

7038 – same as line 7037 plus swap negative sign with space to its right

9289 – first header line missing initial 1 or 2 control character

10234 – mirror temperature has a plus sign after decimal, correct number unknown

14525 – missing text = `EXPLORER VII RADIATION DATA   166 DAYS AFTER LAUNCH`

6920 – contains a line with: `.0000000`

7821, 8399, 11415, 16196 – contain just: `.0000`

8139 – contains a line with: `.  .  .0000`

there are other records that are filled with all blanks.

The all sensors temperature product is in an IBM binary file format, where the 36-bit words are packed into 4½ IEEE 8-bit words. The first record in this product's first data file is 2764 bytes in length, contains no useful information, and can be skipped. Subsequent data records are each of size 1238 bytes the format is shown in the table 3-2 below. File 2 has the following bad records: 546 (1108 bytes), 552-556 (964 bytes) and 557 (1238 bytes). Temperatures are scaled by 100 and stored as integers (questionable values have negative bit set). No geolocation information is in these files.

Table 3-2: Explorer 7 All Sensors Temperature Data Record

Word	Notes	Units
SD SA	Decrement (bits 35-18) = record number part 1 Address (bits 17-0) = record number part 2	
S+1	Days since launch (Oct. 13, 1959 = 0)	
S+2	Hour (time is for first non-zero black-hemisphere 1 sensor)	
S+3	Minute	
S+4	Second	
S+5	Tracking station code	
S+6	unused	
1	High frequency reference	Kelvin x 100 (985.0 ± 15.0)
2	Black hemisphere (1)	Kelvin x 100
3	White	Kelvin x 100
4	Tabor	Kelvin x 100
5	Black hemisphere (2)	Kelvin x 100
6	Gold	Kelvin x 100
7	Black sphere	Kelvin x 100
8	ABMA 1 = Skin temperature	Kelvin x 100
9 – 14	See 2 – 7	
15	ABMA 2 = Solar cell temperature	Kelvin x 100
16 – 21	See 2 – 7	
22	ABMA 3 = Transmitter temperature	Kelvin x 100
23 – 28	See 2 – 7	
29	ABMA 4 = Battery temperature	Kelvin x 100
30 – 35	See 2 – 7	
36	ABMA 5 = Geiger tube temperature	Kelvin x 100
37 – 42	See 2 – 7	
43	Mirror 1 =	Kelvin x 100
44 – 49	See 2 – 7	
50	Mirror 2 =	Kelvin x 100
51 – 56	See 2 – 7	
57	Low frequency reference	Kelvin x 100 (65.0 ± 2.0)
58 – 63	See 2 – 7	
64 – 126	Repeat 1 – 63	
127 – 189	Repeat 1 – 63	
190 – 252	Repeat 1 – 63	
253 – 266	Repeat 1 – 14	
267	Check sum	

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
WestBounding Coordinate	The westernmost longitude of the bounding rectangle(-180.0 to +180.0)
NorthBounding Coordinate	The northernmost latitude of the bounding rectangle(-90.0 to +90.0)
EastBounding Coordinate	The easternmost longitude of the bounding rectangle(-180.0 to +180.0)
SouthBounding Coordinate	The southernmost latitude of the bounding rectangle(-90.0 to +90.0)
ElapsedDays	Duration in days of data collected.

4. Reading the Data

The white sensor (nighttime) temperature product file is written in IBM EBCDIC text format. Each record is a line of text. You will need to convert the EBCDIC character to ASCII to display.

The all sensor temperature product files are written in a IBM 36-bit binary record format. Each 36-bit word is packed into 4½ IEEE 8-bit byte words. Using the record format specification in the section above, users can write software to read the data files.

A sample FORTRAN program is included in the Appendix section which will read and print the the data contents, with functions to convert the data stored in their original IBM EBCDIC or 36-bit words.

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 pages provide information about these data products, as well as links to download the data files and relevant documentation:

https://disc.gsfc.nasa.gov/datacollection/EXP7L1TRTALL_001.html
https://disc.gsfc.nasa.gov/datacollection/EXP7L1TRTWHT_001.html

5.3 Direct Download

These data products are available for users to download directly using HTTPS:

https://acdisc.gesdisc.eosdis.nasa.gov/data/Explorer7_TRE_Level1/EXP7L1TRTALL.001/
https://acdisc.gesdisc.eosdisc.nasa.gov/data/Explorer7_TRE_Level1/EXP7L1TRTWHT.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

“Explorer 7 (1959 IOTA 1) Thermal Radiation Experiment Data Users' Note”, NASA/GSFC NSSDC, March 1967

7. Appendices

Acknowledgments

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

Acronyms

ABMA: Army Ballistic Missile Agency
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
L1: Level-1 Data
NASA: National Aeronautics and Space Administration
EBCDIC: Extended Binary-Coded Decimal Interchange Code
QA: Quality Assessment
UT: Universal Time

FORTRAN Code

```
C-----------------------------------------------------------------------
C ^NAME: READ_EXPL7
C   This program reads Explorer 7 Thermal Radiation Experiment files.
C
C   The Explorer 7 Thermal Radiation Experiment files come in two
C   products: temperatures from the white sensor (nighttime) in an IBM
C   EBCDIC text format, and temperatures from all sensors in an IBM
C   binary 36-bit word format. The EBCDIC file data records are 84 bytes
C   and the first 17 records can be ignored. The binary files contain
C   data records of size 1238 bytes, other records can be ignored
C   (note, record 557 has 1238 bytes but is bad and can be ignored too).
C   This program will print the contents of each data file.
C
C ^MAJOR VARIABLES:
C     FNAME  - name of input file
C     IRECSZ - size of record in bytes
C     BUFF   - buffer for data record
C     WD36   - array for 36-bit words
C     IOS    - I/O status number
C
C ^NOTES:
C     Compile: gfortran -o READ_EXPL7.EXE READ_EXPL7.FOR
C
C ^AUTHOR:  James Johnson (James.Johnson@nasa.gov), NASA GES DISC
C
C ^HISTORY: July 29, 2021 - first version
C-----------------------------------------------------------------------
      PROGRAM READ_EXPL7
      CHARACTER FNAME*1024                  ! Filename
      CHARACTER BUFF(4500)                  ! Buffer for data record
      INTEGER*8 WD36(1000)                  ! Array for 36-bit words
      INTEGER*4 IRECSZ                      ! Size of records
      INTEGER*4 IWORD                       ! 4-byte word
      REAL*4 NPACK                          ! 36-bit packing method
      LOGICAL BCD                           ! Flag if EBCDIC (White Sensor) file
      CHARACTER TEMP(4)                     ! Buffer to hold 4-byte word
      EQUIVALENCE (TEMP,IWORD)

C     Get the name of the input data file to read
      WRITE (0, *), 'Enter the name of the input file:'
      READ (5,'(A)') FNAME
      PRINT '("File = ",A160)', 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 N (record number) and IOFF (byte offset in file)
      N=0
      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) TEMP(I)
          IRECSZ = IWORD
        END DO
        IOFF=IOFF+(I-1)
        IF (ISHFT(IWORD,-31) .EQ. 1) THEN                  ! Check Bit 31
          IRECSZ = IAND(IWORD,'7FFFFFFF'Z)
        END IF

C       End-of-File (EOF) mark, break out of do loop
        IF (IRECSZ .EQ. 0) GOTO 90

        IF (N .EQ. 0) THEN                                 ! Check file type
          IF (IRECSZ == 1844) THEN
            BCD = .TRUE.
          ELSE
            BCD = .FALSE.
          END IF
        END IF

C       Next read the data record
        DO I=1,4500
          BUFF(I)=CHAR(0)                                  ! Clear the buffer
        END DO
        DO I=1,IRECSZ
          READ (1, REC=IOFF+I, IOSTAT=IOS) BUFF(I)
          IF (IOS .NE. 0) THEN
            PRINT '("ERROR: BUFF ",I4,X,I4,", IOSTAT: ",I6)', N,I-1,IOS
            IRECSZ = I-1
            GOTO 90
          END IF
        END DO
        IOFF=IOFF+(I-1)
        N=N+1

        IF (BCD) THEN                                      ! White Sensor Record
          IF (IRECSZ .EQ. 84) THEN                         ! Skip first 17
records
            CALL PRTWHT(IRECSZ,BUFF)
          END IF
        ELSE                                               ! All Sensors Record
          NPACK = 4.5
          CALL BLK2WD(BUFF,IRECSZ,NPACK,WD36)
C         Skip bad records (file 1: rec 1; file 2: recs 546, 552-556
          IF (IRECSZ .EQ. 1238) THEN
            IF (ISHFT(WD36(1),-18) .LT. 2**12) THEN        ! bad file 2: rec 557
              CALL PRTALL(INT(IRECSZ/NPACK),WD36)
            END IF
          END IF
        END IF
```

```
        DO I=1,4
          READ (1, REC=IOFF+I, IOSTAT=IOS, ERR=90) TEMP(I)
        END DO
        IOFF=IOFF+(I-1)

      END DO

C     Close the input file
   90 CLOSE(1)
      STOP

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

C-----------------------------------------------------------------------
C     This Subroutine unpacks 36-bit words into 64-bit long long integer
C-----------------------------------------------------------------------
      SUBROUTINE BLK2WD(BUFF, IRECSZ, NPACK, WD36)
      CHARACTER BUFF(4500)                          ! Buffer for record block
      REAL*4 NPACK                                  ! 36-bit packing method
      INTEGER*8 W36BIT                              ! Use 64-bit for 36-bit int
      INTEGER*8 WD36(1000)                          ! Array of 64-bit words

C     Store 36-bit words into 64-bit double long IEEE integers
      DO N=1,1000
        IF (NPACK.EQ.4.5) THEN
          WD36(N) =
     &    W36BIT(BUFF(INT((N-1)*NPACK)+1:INT(N*NPACK+0.5)), N)
        ELSE
          WD36(N) =
     &    W36BIT(BUFF(INT((N-1)*NPACK)+1:INT(N*NPACK)), 0)
        END IF
      END DO

      RETURN
      END

C-----------------------------------------------------------------------
C     This Function will convert data into a 36-bit word
C-----------------------------------------------------------------------
      INTEGER*8 FUNCTION W36BIT(BUFF,N)
      CHARACTER BUFF(8)
      CHARACTER TEMP(8)
      INTEGER*8 WD36
      EQUIVALENCE (TEMP,WD36)

      IF (N.NE.0) THEN
        DO I=1,8
          IF (I.LE.5) THEN
            TEMP(8-I-2) = BUFF(I)                   ! swap the byte order
          ELSE
            TEMP(I) = CHAR(0)
          END IF
        END DO
```

```
        IF (MOD(N,2).EQ.0) THEN
          WD36 = IAND(WD36,'FFFFFFFFF'Z)
        ELSE
          WD36 = ISHFT(WD36,-4)
        END IF
      ELSE
        WD36 = 0
        DO I=1,8
          WD36 = ISHFT(WD36, 6)                          ! shift left 6 bits
          WD36 = IOR(WD36,IAND(ICHAR(BUFF(I)),'3F'Z))    ! remove 2 MSBs
        END DO
      END IF

      RETURN
      END

C-----------------------------------------------------------------------
C     This Subroutine checks if 36-bit word is negative and returns
C     the negative value
C-----------------------------------------------------------------------
      INTEGER*8 FUNCTION ISNEG(WD36)

      INTEGER*8 WD36, WORD

      IF (ISHFT(WD36,-35) .EQ. 1) THEN
        WORD = -1*IAND(WD36,'7FFFFFFFF'Z)
      ELSE
        WORD = WD36
      END IF

      RETURN
      END

C-----------------------------------------------------------------------
C     This Subroutine will print the White Sensor (Nighttime)
C     Temperature records
C-----------------------------------------------------------------------
      SUBROUTINE PRTWHT(ILEN, BUFF)
      CHARACTER BUFF(4500)                          ! Buffer for record block
      CHARACTER ASC*4500                            ! Buffer for ASCII label

      DO I = 1,ILEN
        ASC(I:I) = CHAR(IEBC(ICHAR(BUFF(I))))
      END DO

      PRINT '(A)', ASC(1:ILEN)

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C     This Subroutine will print the All Sensors Temperature records
C-----------------------------------------------------------------------
      SUBROUTINE PRTALL(NWORDS,WD36)
      INTEGER*8 WD36(1000), ISNEG                    ! record (36-bit words)
      INTEGER*4 WORDD, WORDA                         ! decrement/address 1/2
words
      INTEGER*4 S /1/                                ! header initial offset
      CHARACTER*10 REF(9) /"HIGH FREQ","ABMA1","ABMA2","ABMA3","ABMA4",
     +                     "ABMA5","MIRROR1","MIRROR2","LOW FREQ"/

      WORDD = ISHFT(WD36(S),-18)
      WORDA = IAND(WD36(S),'3FFFF'Z)

      PRINT '("RECORD     = ",2(X,I5))', WORDD, WORDA
      PRINT '("DAY        = ",I12)', WD36(S+1)
      PRINT '("HOUR       = ",I12)', WD36(S+2)
      PRINT '("MINUTE     = ",I12)', WD36(S+3)
      PRINT '("SECOND     = ",I12)', WD36(S+4)
      PRINT '("STATION    = ",I12)', WD36(S+5)
      PRINT '("UNUSED     = ",I12)', WD36(S+6)
      IOFF=S+6

      DO I=1,5
        PRINT '("-------------------------")'
        DO J=1,9
          N=(I-1)*63+(J-1)*7+1
          IF (N .EQ. 267) GOTO 10
          PRINT '(A10,X,"= ",F12.2)', REF(J),ISNEG(WD36(N+IOFF))/100.
          PRINT '("BLACKHS1   = ",F12.2)', ISNEG(WD36(N+1+IOFF))/100.
          PRINT '("WHITE      = ",F12.2)', ISNEG(WD36(N+2+IOFF))/100.
          PRINT '("TABOR      = ",F12.2)', ISNEG(WD36(N+3+IOFF))/100.
          PRINT '("BLACKHS2   = ",F12.2)', ISNEG(WD36(N+4+IOFF))/100.
          PRINT '("GOLD       = ",F12.2)', ISNEG(WD36(N+5+IOFF))/100.
          PRINT '("BLACK      = ",F12.2)', ISNEG(WD36(N+6+IOFF))/100.
        END DO
      END DO
10    PRINT '("-------------------------")'
      PRINT '("CHKSUM     = ",I12)', WD36(N+IOFF)
      PRINT '("=========================")'

      RETURN
      END
```

```
C-----------------------------------------------------------------------
C     This Function returns EBCDIC to ASCII character index
C-----------------------------------------------------------------------
      FUNCTION IEBC(I)
      INTEGER EBCTBL(256)

      DATA EBCTBL /
     & 000,001,002,003,026,009,026,127,026,026,026,011,012,013,014,015, ! 0_
     & 016,017,018,019,026,133,008,026,024,025,026,026,028,029,030,031, ! 1_
     & 026,026,026,026,026,010,023,027,026,026,026,026,026,005,006,007, ! 2_
     & 026,026,022,026,026,026,026,004,026,026,026,026,020,021,026,026, ! 3_
     & 032,026,026,026,026,026,026,026,026,026,162,046,060,040,043,124, ! 4_
     & 038,026,026,026,026,026,026,026,026,026,033,036,042,041,059,172, ! 5_
     & 045,047,026,026,026,026,026,026,026,026,166,044,037,095,062,063, ! 6_
     & 026,026,026,026,026,026,026,026,026,096,058,035,064,039,061,034, ! 7_
     & 026,097,098,099,100,101,102,103,104,105,026,026,026,026,026,177, ! 8_
     & 026,106,107,108,109,110,111,112,113,114,026,026,026,026,026,026, ! 9_
     & 026,126,115,116,117,118,119,120,121,122,026,026,026,026,026,026, ! a_
     & 094,026,026,026,026,026,026,026,026,026,091,093,026,026,026,026, ! b_
     & 123,065,066,067,068,069,070,071,072,073,026,026,026,026,026,026, ! c_
     & 125,074,075,076,077,078,079,080,081,082,026,026,026,026,026,026, ! d_
     & 092,026,083,084,085,086,087,088,089,090,026,026,026,026,026,026, ! e_
     & 048,049,050,051,052,053,054,055,056,057,026,026,026,026,026,026/ ! f_
C       _0  _1  _2  _3  _4  _5  _6  _7  _8  _9  _A  _B  _C  _D  _E  _F

      IEBC = EBCTBL(I+1)

      RETURN
      END
```