# FIRE Cirrus 1 Readme

## 1.0 Introduction

This file contains information about the First ISCCP Regional Experiment (FIRE) Cirrus 1 sample read software and data files. There is read software for all FIRE Cirrus 1 data sets. Data formats for these data sets are categorized into three groups: FIRE Standard Data Format (SDF), FIRE modified SDF which contains Surface Radiation Budget (SRB) data, and National Center for Atmospheric Research (NCAR) General Process (GENPRO) format. Among the SDF data sets, the Data Description Record (DDR) section of the header file for the FIRE Cirrus 1 SRB data sets have been modified to include a record prefix subsection. Two aircraft flight data sets (i.e., Kingair and Sabreliner) are in GENPRO format. This readme file also includes the following five sections:

- Section 2.0 - describes all sample C programs (source files).
- Section 3.0 - discusses how to create the executables from the C program(s).
- Section 4.0 - demonstrates how to invoke the run-time executable(s).
- Section 5.0 - provides general information on the data sets.
- Section 6.0 - provides more detailed implementation notes.

If users have questions while using the FIRE Cirrus 1 sample read software, please contact the Langley DAAC User and Data Services (UDS) office. The UDS may be reached by e-mail: userserv@eosdis.larc.nasa.gov, by telephone at (804)864-8656, or by FAX at (804)864-8807. The UDS mailing address is:

Langley DAAC User and Data Services Office
NASA Langley Research Center
Mail Stop 157D
Hampton, VA 23681-0001

## 2.0 Read Software Files

The sample read programs for FIRE Cirrus 1 data sets are developed in the C language. These C programs have been tested on the ‘Sun4 (OS 4.1.3)’ and ‘HP’ UNIX-based workstations. All library functions in the C source codes are tested for ANSI C conformance. A total of three executable files are generated for the FIRE Cirrus 1 data sets. A ‘Makefile’ is included for the convenience of creating the executable files from the C source files. Two files are needed to generate the read software for NCAR GENPRO data: genpro.c and fire_lib.c. Five files are used to generate the read software for FIRE SDF data: firesdf_read.c, fire_sdf.h, fire_lib.c, fsdf_header.c, and fsdf_vtoc.c. Five files are used to generate the read software for the FIRE Cirrus 1 SRB data: fci1srb_read.c, fire_sdf.h, fire_lib.c, fsdf_header.c, and fsdf_vtoc.c. These files and their purposes are listed as follows:

- **Makefile.fci1** - contains a set of commands to build the run-time executables.
- **genpro.c** - reads NCAR GENPRO formatted data, i.e., Kingair and Sabreliner data sets.
- **firesdf_read.c** - reads SDF formatted data.
- **fci1srb_read.c** - reads the modified SDF formatted SRB data.
- **fire_lib.c** - has a collection of library routines that are used by the other C source files.
- **fire_sdf.h** - defines constants and data structures used in the read routines for the FIRE SDF formatted data sets.
- **fsdf_header.c** - parses the FIRE SDF header file and extracts the field definitions of each actual or implicit variable in the data description record (DDR) for both ancillary and observation segments.
- **fsdf_vtoc.c** -reads a record of FIRE SDF Volume Table Of Contents (VTOC).

# 3.0 Creating C Executable Files

The user can have two ways of building the sample read program from the C source files. One is applying the ‘make’ command to the file ‘Makefile’, which defines the C source files listed in Section 2.0. The other is using the C compiler ‘cc’ (or ‘acc’ for Sparc ANSI C compiler). Section 3.1 illustrates the first method, and Section 3.2 illustrates the second method. The first method is simpler to use than the second one. The symbol ‘%’ in this section stands for the UNIX command line prompt.

## 3.1 Create with ‘make’ command

To build the executable files for all FIRE Cirrus 1 data sets, at the command line prompt, type

% **make all**

It will produce three executable files (i.e., fsdf, fci1_srb, and fci1_aircraft) for reading FIRE Cirrus 1 SDF, FIRE Cirrus 1 SRB, and GENPRO data sets. Make sure the make file name is either ‘Makefile’ or ‘makefile.’ If the ‘make’ file has a different name, e.g., Makefile.fci1, type

% **make -f Makefile.fci1 all**

To build one of the three executables for FIRE Cirrus 1 data sets, at the command line prompt, type

% **make *executable_name***

where *executable_name* is fsdf, fci1_srb, or fci1_aircraft. Makefile.fci1 defines the library file, fire_lib.c, and the header file, fire_sdf.h, to be at the same directory as the other C files. If any of these C files are not at the same directory, the definitions in the 'Makefile' should be changed to point to the right directory.

### 3.2 Create with C compiler 'cc'

This method uses either the combination of the library file, fire_lib.c, and genpro.c, or the library file and SDF C source files. The include file, fire_sdf.h, should also be in the same directory. To build the executable file for the aircraft data sets from the genpro.c source file, type

% **cc -o fci1_aircraft genpro.c fire_lib.c**

and, to build the executable for the FIRE SDF data sets, type

% **cc -o fsdf firesdf_read.c fire_lib.c fsdf_header.c fsdf_vtoc.c**

Similarly, to build the executable for the FIRE Cirrus 1 SRB data sets, type

% **cc -o fci1_srb fci1srb_read.c fsdf_header.c fsdf_vtoc.c fire_lib.c**

Note that the details of creating three executable files are handled in the Makefile under the first method.

## 4.0 Invoking Read Software

This section illustrates the invocation of the executable file. Section 4.1 describes how to run the read programs for the FIRE SDF data sets. Section 4.2 describes how to run the read programs for the FIRE Cirrus 1 SRB data sets. Section 4.3 describes how to run the read programs for the two aircraft data sets.

### 4.1 Running Read Program for the FIRE SDF data sets

At this time, the executable file for SDF data sets can run one data file at each invocation. The SDF read program requires three files to execute this program properly. In order to process each data file accurately, the header filename needs to be specified as the first argument, the second argument is the VTOC filename, and finally the third argument is the data filename. For example, if the data file ci1_er2_861013_1102.004 is to be processed, its corresponding header file is ci1_er2_861013_1102.001, and its table of contents file (VTOC) is ci1_er2_861013_1102.002, then type:

% **fsdf ci1_er2_861013_1102.001 ci1_er2_861013_1102.002 ci1_er2_861013_1102.004** fsdf will read the binary data file, **ci1_er2_861013_1102.004,** and send the output to the standard output device. The program will then prompt the user to select which parameters to output to the standard device. The program can also run in a manner so that the output can be sent to a file by typing:

% **fsdf ci1_er2_861013_1102.001 ci1_er2_861013_1102.002 ci1_er2_861013_1102.004** ***num_of params param1 param2 param3 ... param# > any_-file_name***

where *num_of_params* is the number of parameters the user would like to output to the file, and *param1 ... param#* are the actual list of parameter numbers to be written to the file.

See Section 6.0 IMPLEMENTATION NOTES for the SDF data output.

## 4.2 Running Read Program for the FIRE Cirrus 1 SRB data sets

Currently the executable file for these data sets which include SRB_Alaska, SRB_Canada, SRB_Switzerland, and SRB_SouthPole can run one data file at each invocation. In order to process the data file accurately the header file needs to be specified as the first argument, the VTOC filename as the second argument, the third argument is the desired data file, and the fourth argument is the corresponding ancillary file referred to by the data file. For example, if we need to process a data file ci1_srb_sp_861020_25.007, its corresponding header file is ci1_srb_sp_861020_25.001, the corresponding VTOC file is ci1_srb_sp_861020_25.002, then the data file ci1_srb_sp_861020_25.007, and finally the corresponding ancillary file ci1_srb_sp_861020_25.004. Hence type:

% **fci1_srb ci1_srb_sp_861020_25.001 ci1_srb_sp_861020_25.002 ci1_srb_sp_861020_25.007 ci1_srb_sp_861020_25.004**

fci1_srb will read binary data from **ci1_srb_sp_861020_25.007** and create an output file called ci1_srb_sp_861020_25.007**.**OBS. Please note that all SRB_Alaska and SRB_-SourthPole data sets have a single ancillary file hence the ancillary file referred by the data file is always filename.004 (e.g. ci1_srb_sp_861020_25.004). But SRB_Canada and SRB_Switzerland have multiple ancillary files. Therefore, the ancillary file referred by the data file can be either filename.004 or filename.005.

## 4.3 Running Read Program for the Aircraft data sets

Currently, the executable files for the Kingair and Sabreliner aircraft data sets can run one data file at each invocation. The aircraft data files are organized in pairs, i.e., one header file has a corresponding flight data file. In order to process the data file accurately, the header file needs to be specified as an argument preceding the data file argument. For example, if the Kingair data file ci1_ka_flt01_861014.dat is to be processed, and its corresponding header file is ci1_ka_flt01_861014.hdr, then type

% **fci1_aircraft ci1_ka_flt01_861014.hdr ci1_ka_flt01_861014.dat**

will read the binary data in the ci1_ka_flt01_861014.dat and send the output to the screen. The output can be directed to a file by typing

% **fci1_aircraft ci1_ka_flt01_861014.hdr ci1_ka_flt01_861014.dat >** ***any_file_name***

See Section 6.0 IMPLEMENTATION NOTES for the aircraft data output.

# 5.0 Data Sets Information

There are six sections describing each data set group. Each section is further divided into subsection(s) that contain descriptions of each data set. The high-level data set descriptions, the data files, the variables in the data files, and the contacts for data production are given under each data set. Some data sets contain the instrument information.

## 5.1 FIRE SDF Data Sets

These data sets were written in the FIRE Standard Data Format (SDF). For a complete description of this format, please refer to the "Standard Data Format for FIRE" written by NASA's Climate Data System (NCDS). Most data sets did not follow this format exactly as stated in this paper: however, these differences are noted in the read program. When a user orders a SDF data set, he will receive a header file, VTOC file, and an ASCII representation of the first data file. (The first data file ends in .004.) All data experiments contain a fill value integer of 2147483647. In all cases, this value has been ignored.

In most of the experiments, you will notice in the comments that the Greenwich Mean Time (GMT) begin and end dates for each binary data file contains blanks instead of using zeroes. The read program will inform the user to be aware of this when working on the dates for each ASCII header record at the top of each data file. There were also some GMT dates that were not written in the correct columns. (The actual format of the ASCII header record is explained in the FIRE Standard Data Format User's Guide.) This causes an error to appear while checking the GMT dates. You will notice an error message that states that either the longitude values or latitude values are out of range. Please ignore these messages unless the experiment comments located below actually states that these values are out of range. In a couple of instances, this is true.

### 5.1.1 Meteorological and Radiometric Data from the NCAR Sabreliner

**FILE NAMING CONVENTION:**

ci1_sabr_861013_1102.001 (header file)

ci1_sabr_861013_1102.002 (VTOC file)

ci1_sabr_861013_1102.003 (ASCII representation of the first data file, .004)

ci1_sabr_861013_1102.004 (first binary data file)

This data set contains 25 binary data files.

**VARIABLES:**

All records and parameters within this data set have been defined, including their minimum and maximum values, in the header file, ci1_sabr_861013_1102.001.

**NOTES ON DATA:**

After testing this data set, we have found that the southern and northern latitude values are written in reverse order. This information is written in the VTOC file as well as in each of the data record files. The data producer recommends that the user is made aware of this finding.

The fill value number is the maximum 4-byte integer word size of 2147483647.

There are two full records of (176*2) ASCII header information in each data record file.

There were no problems with values being out of the minimum/maximum range.

**INSTRUMENTATION:**

This information is provided within the header file for this experiment.

**CONTACTS FOR DATA PRODUCTION INFORMATION:**

Paul F. Hein

Department of Atmospheric Science

Colorado State University

Fort Collins, CO 80523

Phone: (303) 491-8714

FAX: (303) 491-8449

e-mail: hein@trueno.atmos.colostate.edu

### 5.1.2 NASA ER2 Radiometer Measurements

**FILE NAMING CONVENTION:**

ci1_er2_861013_1102.001 (header file)

ci1_er2_861013_1102.002 (VTOC file)

ci1_er2_861013_1102.003 (ASCII representation of the first data file, .004)

ci1_er2_861013_1102.004 (first binary data file)

This data set contains 22 binary data files.

**VARIABLES:**

All records and parameters within this data set have been defined, including their minimum and maximum values, in the header file, ci1_er2_861013_1102.001.

**NOTES ON DATA:**

In the header file, the record name is the same for both records that are defined. Both records are declared as having 32 bytes. The first record declared only uses 28 bytes and the other four bytes are padded with zeroes. In the second record, only 8 bytes are defined and the last twenty bytes are padded with zeroes. The first record that is listed in the header file starts with .004 and following that file is the second record in file .005. This continues throughout all of the data files. The DAAC recommends that when you are testing/using these data, please break up the header file to have only one record defined in the file that you plan to work with (since same record name). Just make sure that all files with the even ending number (in the original file name) use the first record and all files with the odd ending file number (in the original file name) use the second record.

The fill value number that we have been working with is the maximum 4-byte integer word size of 2147483647.

The ASCII header record in each data record file was set to 160 bytes.

A few GMT dates located within the ASCII header record in every binary data file had blanks instead of zeroes within the dates.

There were no problems with values being out of the minimum/maximum range.

**INSTRUMENTATION:**

This information is provided within the header file for this experiment.

**CONTACTS FOR DATA PRODUCTION INFORMATION:**

Phillip Hammer

Phone: (415) 604-3383

e-mail: hammer@rad.arc.nasa.gov

### 5.1.3 NWS Rawinsonde Observations

**FILE NAMING CONVENTION:**

ci1_raob_861013_1102.001 (header file)

ci1_raob_861013_1102.002 (VTOC file)

ci1_raob_861013_1102.003 (ASCII representation of the first data file, .004)

ci1_raob_861013_1102.004 (first binary data file)

This experiment contains 553 binary data files.

**VARIABLES:**

All records and parameters within this data set have been defined, including their minimum and maximum values, in the header file, ci1_raob_861013_1102.001.

**NOTES ON DATA:**

The header file has been modified to reflect the changed record names in each file. However, the same parameters exist for each record. The reason for changing the record names is the external read header file reads a specific record name followed by the parameters defined within each record and prior to the change, the file did not possess this information.

The read program has been modified to compute the WIND SPEEDSHEAR parameter in each record. The reason is that if you read the integer value and convert the value to a real value, the new result will always be out of the minimum/maximum range as defined within the header file. The data producer recommended a new formula to convert the integer value that was read in and converted to a real value, the real value must be divided by 2097152.0. Finally, 400.0 must then be subtracted from this new result to get the proper value within the minimum/maximum range. This formula has been implemented in the read program.

The fill value number for this data set is the maximum 4-byte integer word size of 2147483647.

There are two full records of (112*2) ASCII header information in each data record file.

There were no problems with values being out of the minimum/maximum range.

**INSTRUMENTATION:**

This information is provided within the header file for this data set.

**CONTACTS FOR DATA PRODUCTION INFORMATION:**

Dr. David O'C. Starr

Code 613

NASA Goddard Space Flight Center

Greenbelt, MD 80523

Phone: (301) 491-8714
FAX: (301) 286-1759
e-mail: starr@climate.gsfc.nasa.gov

### 5.1.4 Temperature Soundings from TOVS

**FILE NAMING CONVENTION:**

ci1_tovs_861014_1102.001 (header file)
ci1_tovs_861014_1102.002 (VTOC file)
ci1_tovs_861014_1102.003 (ASCII representation of the first data file, .004)
ci1_tovs_861014_1102.004 (first binary data file)

This data set contains 76 binary data files.

**VARIABLES:**

All records and parameters within this data set have been defined, including their minimum and maximum values, in the header file, ci1_tovs_861014_1102.001.

**NOTES ON DATA:**

Modifications were made to the header file at the data producer's request. (1) The record name was missing from the record definition. This has been included. (2) The word Description was misspelled in the section called Instrument Description - the letter p was missing. (3) The parameter Time had a minimum value of 1. This value has been changed to 0. (4) The parameter HHMM has been changed to HHMMSS to reflect the data. The minimum value for this parameter has been changed from 1 to 0. The maximum value for this parameter has been changed to 240000.

The fill value number for this data set is the maximum 4-byte integer word size of 2147483647.

A few GMT dates located within the ASCII header record in every binary data file had blanks instead of zeroes within the dates.

There were no problems with values being out of the minimum/maximum range.

**INSTRUMENTATION:**

This information is provided within the header file for this data set.

**CONTACTS FOR DATA PRODUCTION INFORMATION:**

Don Wylie

Space Science and Engineering Center
University of Wisconsin-Madison
1225 West Dayton Street
Madison, WI 53706
Phone: (608) 263-7458
e-mail: donw@ssecmail.ssec.wisc.edu

### 5.1.5 NASA ER2 Lidar Measurements

**FILE NAMING CONVENTION:**

ci1_spin_861013_1102.001 (header file)
ci1_spin_861013_1102.002 (VTOC file)
ci1_spin_861013_1102.003 (ASCII representation of the first data file, .004)
ci1_spin_861013_1102.004 (first binary data file)

This data set contains 114 binary data files.

**VARIABLES:**

All records and parameters within this data set have been defined, including their minimum and maximum values, in the header file, ci1_spin_861013_1102.001.

**NOTES ON DATA:**

The header file states that there are 80 bytes in each data record. The last four bytes in each record are padded with zeroes.

The undetected data value has been flagged with a -9.9.

An integer value appears at the end of the file. The data producer is not sure why there are a few records of this same integer appearing in every data file. The data producer has asked that we ignore this value. All data looks good once this value was ignored. This integer value is 538976288. This sample read program has been written to ignore these values.

The fill value number for this data set is the maximum 4-byte integer word size of 2147483647.

The ASCII header record contains 160 bytes; however, the spacing is not the same as what has been defined in the SDF document.

The GMT date within the ASCII header record of each data file contains blanks instead of zeroes.

There were no problems with values being out of the minimum/maximum range.

**INSTRUMENTATION:**

This information is provided within the header file for this data set.

**CONTACTS FOR DATA PRODUCTION INFORMATION:**

Dennis Hlavka

Code 617

Room 315 Building 22

NASA Goddard Space Flight Center

Greenbelt, MD 20771

Phone: (301) 286-5578

e-mail: sgdlh@virl.gsfc.nasa.gov

## 5.2 FIRE Cirrus 1 SRB Data Sets

Data obtained during the FIRE Cirrus 1 SRB experiment were classified on the basis of region. The above data set consists of four region based directories: (1) SRB_SouthPole, (2) SRB_Alaska, (3) SRB_Canada, & (4) SRB_Switzerland.

File No. .001: Header file in character formation (ASCII).

File No. .002: Volume Table of Contents (VTOC) in ASCII. Brief description of files beginning with the ancillary data file(s), the fourth (and fifth, ...).

File No. .003: Test Data in ASCII. Physical record length is 6400 bytes. Logical record length is 80 bytes. Each logical record is an expanded "pixel packet," consisting of the data areas logical record (RSIZ) bytes of the original file per "Test Data" logical record. Note that this file represents the complete contents of the first "observation data" file, NOT of the first "data" file (the ancillary file).

File No. .004: Ancillary File (Lat/Lon grid) in scaled INTEGER*2 format. Record prefix length is RPFX bytes. Logical record length is RSIZ bytes. There are a number of records (NREC) logical records, or "pixels," per physical record. Physical record length is thus (RPFX + NREC * RSIZ) bytes. The LAT/LON quantities are to be used as a look up table to convert any (pixel, scanline) coordinate pair in each of the data files into a pair of LAT/LON coordinates. If there is more than one ancillary data files in a directory, then the second ancillary file will be the fifth physical file.

File No. x through last where “x” = (3 + number of ancillary files): Observation data files are all in INTEGER*1 format. Each physical record is composed of a “record prefix area” and a “record data area.” The record prefix contains the following information:

FNUM (I*4) - Data file number.

RNUM (I*4) - Data record sequence number within the file.

CKSM (I*4) - Checksum (sum of all values within the DATA area that follows the PREFIX area).

RSIZ (I*4) - The number of logical records in the DATA area.

RPFX (I*4) - The size in bytes of the PREFIX area.

RPFX can be obtained from the header file (DDR) or from the PREFIX area itself.

Physical record length is (RPFX + NREC * RSIZ) bytes. It contains various flags and physical quantities in coded format, obtained from ISCCP analysis of B3 radiance data.

The directories, SRB_SouthPole and SRB_Alaska, have single ancillary files hence the “observation data file” starts from the fifth physical file. The directories, SRB_Canada and SRB_Switzerland, have multiple ancillary files hence the “observation data file” is the sixth physical file in the directory.

The granule name consist of the region names and the starting and ending dates of data collection (i.e., ci1_srb_sp_yymmdd_dd) (SRB_SouthPole). Each region is named ci1_srb_region_yymmdd_dd where region can be “sp” (SouthPole), “ak” (Alaska), “ch” (Switzerland), or “ca” (Canada). The yymmdd indicates the data year, the data month in numerical form, and dd represents the begin day of the data collection. The last dd represents the ending day of the data collection.

**VARIABLES:**

Each of the observation data files in FIRE Cirrus 1 SRB contain 24 variables. Each variable has been defined as a one byte unsigned integer. Two variables (Lat/Lon) are stored in each ancillary data files each in INTEGER*2 format. In order to scale the data so they are 1-byte, 2-bytes, or 4-bytes positive integers the following equation is used:

$$Q = (R - A) \; * \; (2**(b - N))$$

where R is the actual (real) data value, b=7 for 1 byte integers, b=15 for 2 byte integers, and b=31 for 4 byte integers and Q is rounded to a positive integer. All records and parameters within each record have been defined including their minimum and maximum values in the header file filename.001.

**NOTES ON DATA:**

Questions about instrumentation and specific data parameters (including their derivation utilization and units) should be directed to Goddard Institute of Space Studies (GISS). There are small discrepancies pertaining to the SRB data set. Some of the data values from data files and ancillary files were out of range when compared against the ranges provided by the VTOC, and the maximum and minimum values from the Header files. The data producers response to these discrepancies was "What is in VTOC is the definition of the region, what is in the Ancillary files is a subset of scan lines falling in that region. A given scan line may have some pixels with lat/lon outside the region."

## 5.3 FIRE Cirrus 1 Aircraft Data Sets

Aircraft data obtained during the FIRE Cirrus 1 experiment were classified according to the aircraft platform from which the measurements were taken. Two aircraft data sets are described here: NCAR Kingair, and NCAR Sabreliner. Although they have different sets of variables, they are generated in the same format. Thus, one read routine is used to extract data from both data sets.

### 5.3.1 KINGAIR & SABRELINER

The Kingair and Sabreliner data files from 1986 FIRE Cirrus 1 are provided by the National Center for Atmospheric Research (NCAR) in General Processor (GENPRO) format (binary positive integers). The Kingair and Sabreliner data files are organized in pairs for an individual flight, i.e., header #1, flight #1 data, header #2, flight #2 data, etc... Header files are in ASCII format, and flight data are in binary format.

**FILES:**

Kingair data set has 15 header-data-file pairs for 13 flights because flight 7 has 3 header-data-file pairs. The Sabreliner data set has 18 header-data-file pairs for 18 flights. The granule names for FIRE Cirrus 1 Kingair and Sabreliner data are the flight names. Each flight is named ci1_aircraft_flt#_flightdate, where "aircraft" is either 'ka' or 'sa' to mean Kingair or Sabreliner, and "flightdate" has the format of yymmdd, e.g., ci1_ka_flt01_861014, and ci1_sa_flt02a_861015. When the data from the entire flight is broken into two or more flights, they are labelled a, b, ... after the flight number. The flight number ranges from 01 to 14 for the Kingair. Flight 3 does not exist and flight 7 is numbered 7a, 7b, and 7c. The flight numbers range from 01 to 18 for the Sabreliner. Flight 2, 5, 7, 10, 12, 13, and 14 have a and b. The header files are named granule_name.hdr, and the data files are named granule_name.dat for Sabreliner data. The header files are named granule_name.hdr, and the data files are named granule.dat for Kingair data.

**VARIABLES:**

Each of the 15 FIRE Cirrus 1 Kingair data files has 161 variables, and each of 18 FIRE Cirrus 1 Sabreliner data files has 106 variables. Each variable is defined as a 4-byte unsigned integer. The unsigned integer values have to be converted to real numbers for correct reading. Usually the variables are sampled at 1 Hz (per genpro cycle), but some variables are sampled at a higher rate. The sample read routine automatically retrieves the information for all defined parameters, and interprets the binary data accordingly.

**NOTES ON DATA:**

Questions about instrumentation and specific data parameters (including their derivation, utilization and units) should be directed to NCAR.

During the process of ingesting FIRE Cirrus 1 data sets, some questions regarding header files and data files arose. 1.) There are small discrepancies between the header beginning and ending flight time specification, and the actual data beginning and ending times. And, in FIRE Cirrus 1 Kingair data, except for data file ci1_ka_fltxx_yymmdd, all other data files have negative values of -1000 for all variables through the last few records. As to FIRE Cirrus 1 Sabreliner data, except for flight 4 and 9, all other data files also have the same negative values (-1000) for all variables through the last few records. 2.) There are no apparent time variables (HR, MIN, SEC) in the Kingair data as there were in the Sabreliner data. 3.) There are three sets of latitude/longitude variables. Larry Miloshevich from NCAR responded the above questions as follows:

The time values were incorrect (negative) for the last few seconds of each flight, and there were beginning time and ending time discrepancies of a few seconds between the times listed in the file headers and the first or last times of the initial and final data records. Although I can't give you a certain cause for these two discrepancies, the bottom line is that they are profoundly unimportant. My guess is that they are due to effects of turning the data system on and off, since the header time interval for the flight is slightly longer than the interval of data collection (i.e., perhaps time marks are recorded for the header when the data collection system is turned on or off, but if it takes a couple of seconds for the system to begin or quit taking data then the data collection interval will be slightly shorter). Also, my guess is that the negative time values at the end of files might be due to flushing data buffers when the system is shut down. In any case, the best advice for users of this data would be: 1) use the time values associated with the data records and ignore the discrepancy with the header files, and 2) eliminate or otherwise don't use the ending records with negative time values, since other data in those records may be untrustworthy.

Regarding time variables in Kingair data: There are indeed times available for the Kingair data files. The variable TPTIME is time in "seconds after midnight." A program is used to convert this number into a six digit "hours minutes seconds", or HHMMSS, just like the Sabreliner records.

Regarding the lat/lon position coordinates: there are three sets of position coordinates recorded: lat/lon from the INS (Inertial Navigation System, variables called ALAT and ALON), lat/lon from LORAN (an external aircraft navigation system, variables called CLAT and CLON), and distances in km east or north from the starting position, which are calculated from one of the above lat/lon measurements and give relative rather than absolute aircraft position (variables called DNI and DEI). I and others have used with reliability the latter relative position measurements, and if individual researchers need lat/lon coordinates they can be calculated from DNI and DEI relative to the known lat/lon coordinates of the starting position (i.e., the airport). Again, flight tracks and other uses of position measurement are known to be reliable using DNI and DEI, and absolute lat/lon coordinates can be calculated from these if CLAT and CLON (the more reliable of the two lat/lon measurements) are thought to be suspect.

**CONTACTS FOR DATA PRODUCTION INFORMATION**

Andy Heymsfield

Phone: (303) 497-8181

e-mail: heyms@mmm.ncar.edu

# 6.0 Implementation Notes

## 6.1 FIRE SDF program

The fsdf_header.c program builds the data structure for the variables defined in the Variable Description (VD) record. When the variable values are to be displayed as octal, or hexadecimal format, the scaling constant 'b' of the conversion function defined in the VD record is stored as 0 for the convenience of coding (the header file is not changed). It is assumed these variables require no conversion from integer to floating point values.

FIRE SDF rules recommend that the bad or missing data to be represented by the largest possible integer value for each data type, but some data sets use different values as the fill value. This FIRE Cirrus 1 SDF sample read program knows what these fill values are and they are identified in the code.

This standard read program is capable of reading a SDF file that is available at the Langley DAAC. Because of the modifications made in this read program to handle the SDF data files, this program could not read any other SDF files not from the Langley DAAC.

This read software reads in all of the information defined in the header file such as the min/max ranges for every parameter, the values are needed to convert the value to a real number, as well as how many parameters are in each record defined. The program prints these parameters onto the screen and the user is able to select those parameters he wishes to print. Currently, the values will be printed out to the screen. The user may redirect his output to another file or modify the program to write out the data in a different presentation or to a file.

## 6.2 SRB program

The SRB read program is capable of reading in a FIRE CI1 SRB data file available at Langley DAAC. This read software reads in all the information from the header file, e.g. maximum and minimum values for each parameter, the values required to convert the data value to a real number, and number of parameters in each record defined. The ASCII header record contains 160 bytes in both the observation file and the ancillary file. No problems were encountered with data values being out of the minimum/maximum range.

## 6.3 GENPRO program

The generic read program (genpro.c) for the FIRE aircraft flight data extracts the flight date, flight start time, and flight end time from the header file (not from the data file) because FIRE Cirrus 1 Kingair flight data files do not have the time variables (HR, MIN, SEC) defined like other aircraft data. However, the minimum and maximum INS (Inertial Navigation System) latitude and longitude values are obtained from the data file. The total number of records in the specified data file is printed out at the end of the output.

Because of the huge amount of data in the data file, the read program will only print out the selected 7 variables (1st, 2nd, 3rd, 4th, 5th, 6th, and 7th variables, in that order) of all records in tabular format. The "printf" statement in the C source file works as a demonstration of printing out values of the desired variables. If these data values are to be printed out, users should re-compile the genpro.c program with a "-DDEBUG" preprocessor option and rerun the executable. The minus D option will produce the values of all seven variables for all records in the specified data file.

# APPENDIX

## 1.0 Output from running ‘make all’ on HP workstation

**HP% make -f Makefile.fci1 all**

```
cc -O -Aa  -I.   -c  fire_lib.c

rm -f ./lib_fire.a

ar rlv ./lib_fire.a ‘lorder fire_lib.o | tsort‘

a - fire_lib.o

ar: creating ./lib_fire.a

ranlib ./lib_fire.a

ranlib: ar already did it for you, see ar(1).

rm -f fci1_aircraft

cc -O -Aa  -I.    -o fci1_aircraft genpro.c ./lib_fire.a

cc -O -Aa  -I.   -c  firesdf_read.c

cc -O -Aa  -I.   -c  fsdf_header.c

cc -O -Aa  -I.   -c  fsdf_vtoc.c

rm -f fsdf

cc -O -Aa  -I.    -o fsdf firesdf_read.o  fsdf_header.o  fsdf_vtoc.o ./lib_fire.a -lm

rm -f fci1_srb

cc -O -Aa  -I.    -DSRB -o fci1_srb fci1srb_read.c  fsdf_header.c  fsdf_vtoc.c ./lib_fire.a -
lm

fci1srb_read.c:

fsdf_header.c:

fsdf_vtoc.c:
```

# 2.0 Output from running ‘make all’ on Sun workstation

**SUNOS% make -f Makefile.fci1 all**

acc -O   -I.  -sun4 -c  fire_lib.c

rm -f lib_fire.a

ar rlv lib_fire.a ‘lorder fire_lib.o | tsort‘

a - fire_lib.o

ar: creating lib_fire.a

ranlib lib_fire.a

rm -f fci1_aircraft

acc -O   -I.   -sun4 -o fci1_aircraft genpro.c ./lib_fire.a

acc -O   -I.  -sun4 -c  firesdf_read.c

acc -O   -I.  -sun4 -c  fsdf_header.c

acc -O   -I.  -sun4 -c  fsdf_vtoc.c

rm -f fsdf

acc -O   -I.   -sun4 -o fsdf firesdf_read.o  fsdf_header.o  fsdf_vtoc.o ./lib_fire.a /usr/lib/libm.a

rm -f fci1_srb

acc -O   -I.   -sun4 -DSRB -o fci1_srb fci1srb_read.c  fsdf_header.c  fsdf_vtoc.c ./lib_fire.a /usr/lib/libm.a

fci1srb_read.c:

fsdf_header.c:

fsdf_vtoc.c:

Linking: