NOAA Technical Memorandum ERL SEL-86

TIROS/NOAA SATELLITE SPACE ENVIRONMENT MONITOR DATA ARCHIVE DOCUMENTATION: 1995 UPDATE

V. J. Raben
D. S. Evans
H. H. Sauer
S. R. Sahm
M. Huynh

Space Environment Laboratory
Boulder, Colorado
February 1995

UNITED STATES DEPARTMENT OF COMMERCE

Ronald H. Brown
Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

D. JAMES BAKER
Under Secretary for Oceans and Atmosphere/Administrator

Environmental Research Laboratories

James L. Rasmussen
Director

NOTICE

Mention of a commercial company or product does not constitute an endorsement by NOAA Environmental Research Laboratories. Use for publicity or advertising purposes of information from this publication concerning proprietary products or the tests of such products is not authorized.

For sale by the National Technical Information Service, 5285 Port Royal Road
Springfield, VA 22161

CONTENTS

TIROS/NOAA Satellite Space Environment Monitor Data Archive Documentation: 1995 Update

V. J. Raben, D. S. Evans, H. H. Sauer, S. R. Sahm, M. Huynh

ABSTRACT

TIROS/NOAA satellite archive tapes containing data obtained with the Medium Energy Proton and Electron Detector (MEPED), High Energy Proton and Alpha Particle Detector (HEPAD), and Total Energy Detector (TED) are described. Descriptions of the data include orbital and housekeeping details and the information needed to decode and understand the data. Specifications of the data channels are supplied, as is the timing information needed to convert the data to usable information. Description of the archive tape format gives the information needed to read the tape and unpack the data. Appendices supply the retrieval routines used by the Space Environment Services Center in Boulder.

1.0. Introduction

The TIROS/NOAA (Television and Infrared Observation Satellite/National Oceanic and Atmospheric Administration) satellites carry a set of instruments to detect and monitor the influx of ions and electrons into the upper atmosphere as a result of solar and magnetospheric activity. This set of instruments is called the Space Environment Monitor (SEM). SEM data are received in near–real time at the Space Environment Services Center (SESC) of the Space Environment Laboratory in Boulder, Colorado. The data are used operationally by SESC and are also archived on 3480 magnetic tape. Tape copies can be obtained from the following source:

National Oceanic & Atmospheric Administration
National Environmental Satellite, Data, and Information Service
National Geophysical Data Center E/GC2
325 Broadway
Boulder, Colorado 80303, U.S.A.

The TIROS/NOAA archive tapes contain orbital and housekeeping information, as well as data from the three SEM instruments:

1. The MEPED: Medium Energy Proton and Electron Detector.
2. The HEPAD: High Energy Proton and Alpha Particle Detector (none after NOAA-7).
3. The TED: Total Energy Detector.

Seven TIROS/NOAA satellites have been launched. The lifetime of each satellite is about 2 years; at most, two satellites are operational at any time. Figure 1.1 shows the location of the SEM instruments on the spacecraft.

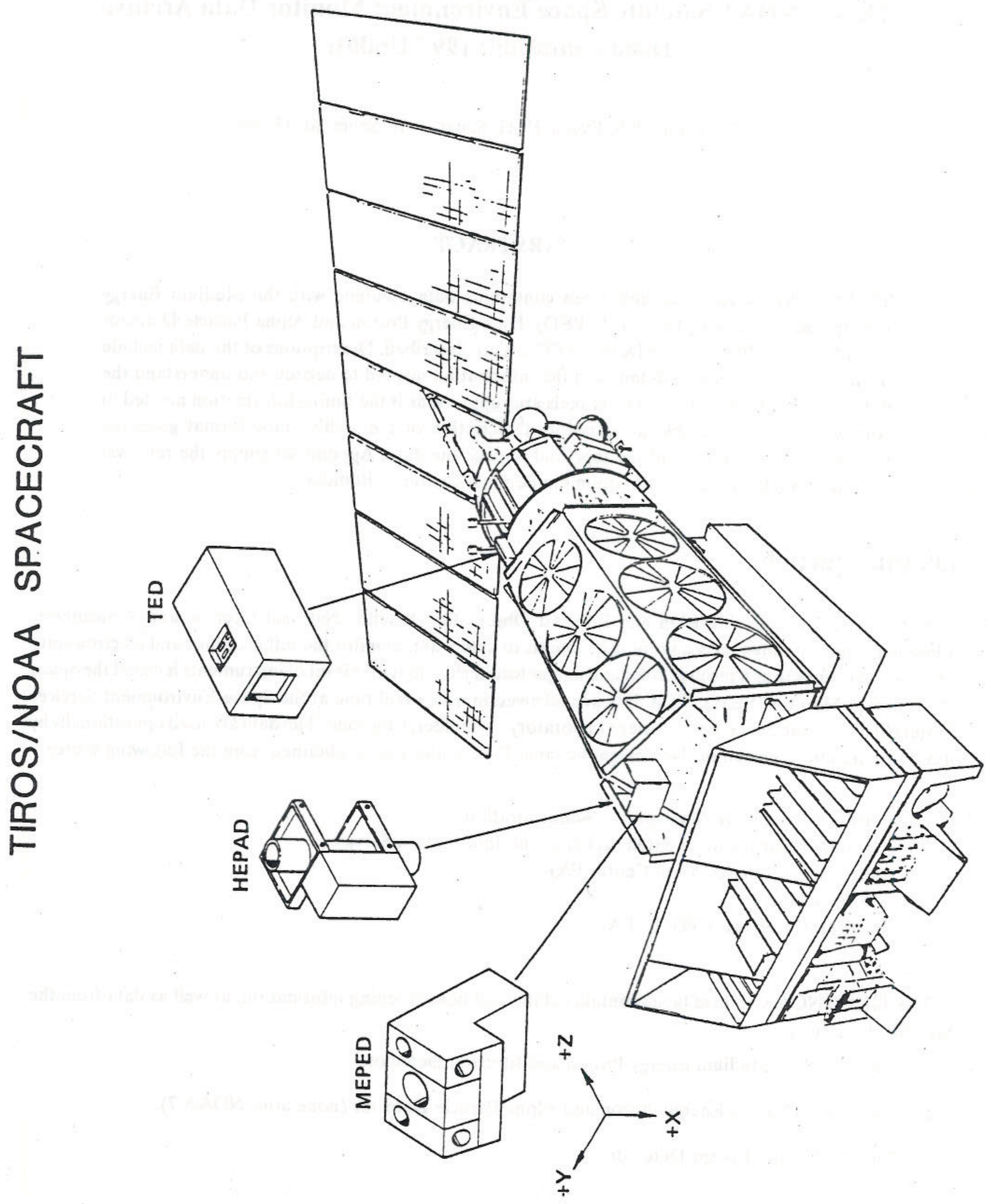


Figure 1.1. SEM instruments on TIROS/NOAA spacecraft.

Table 1.1. shows the data available on the TIROS/NOAA Archive tapes. Archive tapes are created within 2 weeks of data collection.

Table 1.1. Data on TIROS/NOAA Archive Tapes

Satellite	Data on Archive Tapes		
TIROS–N	02 November 1978	–	27 February 1981
NOAA–6	28 June 1979	–	09 May 1983
	01 July 1984	–	30 June 1985
	15 October 1985	–	18 November 1986
NOAA–7	11 July 1981	–	10 February 1985
	MEPED and HEPAD turned off 01 April 1982		
NOAA–8	09 May 1983	–	14 June 1984
	01 July 1985	–	14 October 1985
	No HEPAD instrument		
NOAA–10	11 October 1986	–	31 August 1991
	No HEPAD instrument		
NOAA–12	14 May 1991	–	current
	No HEPAD instrument		
NOAA–14	29 Dec 1994	–	current
	No HEPAD instrument.		

This technical memorandum is designed to assist the user in reading and decoding the NOAA Satellite Archive Tapes and in understanding the information contained in the SEM data. It includes descriptions of the instruments and detailed specifications of the data channels, and gives timing information. These specifications are necessary to convert the data to usable information.

Since its original publication in 1985 this technical memorandum has been updated from time to time as circumstances warranted. This 1995 update is occasioned by a change in the computer system used by the Space Environment Laboratory and the necessity to reformat all historical TIROS/NOAA SEM data onto 3480 tape cartridges (from the original 9-track, 1600-bpi magnetic tape reels). Because all historical data had to be reprocessed, this changeover permitted several errors that had been present in the original data to be corrected before the data were written to the 3480 cartridges. The errors that have been rectified include (1) the miscalculation of auroral particle energy fluxes deposited into the atmosphere whenever the satellite was in the southern hemisphere (an error in archive data prior to 1984) and (2) the miscalculation of the particle pitch angles viewed by the MEPED detectors when the satellite was southbound (an error in archive data prior to NOAA-10, which was launched in late 1986).

The pitch angle convention has now been made uniform for both TED and MEPED data. This convention is that the pitch angle refers to the angle between the particle's velocity and the local magnetic field vector, so that in the northern hemisphere precipitating particles possess pitch angles between 0 degrees and 90 degrees and in the southern hemisphere such particles possess pitch angles between 90 degrees and 180 degrees.

Data quality control also was much improved in the reformatted data; in particular, bad or questionable data within a 2-second data record are now keyed by inserting a value of 1000 for the corresponding total auroral particle energy flux values.

The SEM instrument descriptions were written by the primary investigators. For more information about the instruments, data, current uses, and published papers, contact Dr. D. S. Evans (about the TED), Dr. M. V. Codrescu (about the MEPED), or Dr. H. H. Sauer (about the MEPED and HEPAD), at the National Oceanic & Atmospheric Administration, Space Environment Laboratory R/E/SE, 325 Broadway, Boulder CO 80303. The SEM instruments and the raw telemetry format (with housekeeping words) are described in detail in the NOAA Technical Memorandum "The TIROS−N/NOAA A−J Space Environment Monitor Subsystem" (Seale, R. A., and R. H. Bushnell, 1987, NOAA Tech. Memo. ERL SEL−75, NOAA Environmental Research Laboratories, Boulder, Colo.)

2.0. MEPED Instrument Description

The MEPED (Medium Energy Proton and Electron Detector) is that portion of the SEM designed to measure the flux of protons (ions) and electrons mirroring above, and precipitating into, the high-latitude atmosphere. Each MEPED consists of two sensor assemblies: the directional (telescope) particle detectors and the omnidirectional proton detectors.

The telescopes are mounted in two pairs, one of each pair detecting electrons, the other detecting protons (and heavier ions). One pair of detectors is mounted to view outward along the Earth-satellite radial vector zenith. At geomagnetic latitudes greater than 30 degrees, these detectors view charged particles that are in the atmospheric loss cone and will enter the atmosphere. The other detector pair is mounted to view at about 80 degrees to the first, and for magnetic latitudes greater than 30 degrees will measure particles that have pitch angles near 90 degrees (i.e., particles that are outside the loss cone and are trapped). For convenience these two detector telescopes are identified with the suffix 0 and 90. The local pitch angles of the particles observed by these two pairs of directional detectors at any point in the orbit are calculated using a model magnetic field developed at the National Space Science Data Center (Stassinopoulos and Mead, 1972). The pitch angles are included in the archive tape record as part of the header information.

The electron detector is a thin (700 μm) 25 mm ^{2} solid-state detector covered by 0.51-μm-thick nickel foil (0.70 μm in the case of TIROS-N), that suppresses detector response to photons and reduces pulse pile-up caused by incident low-energy electrons or ions. Electronic pulse-height discrimination is used to select pulses due to incident electrons of nominal energies greater than 30 keV, 100 keV, and 300 keV (taking into account a nominal 5 keV energy loss as the electron passes through the foil). The contaminant response to protons that deposit more than 1 MeV in the detector is eliminated electronically. The detectors are, however, sensitive to protons between about 135 keV and 1 MeV. Data from the directional proton detectors may be used to correct for this effect.

The proton (ion) detector within each telescope pair is a two-element, solid-state detector telescope. The front element has an effective area of 25 mm ^{2} and thickness of 200 μm. The back element has an effective area of 50 mm ^{2} and a thickness of 200 μm. A 2500-gauss magnet is mounted across the input aperture of this detector assembly to prevent any electrons of energies less than 1.5 MeV from reaching the detectors. The front face of the front detector of the telescope is coated with an aluminum layer 18 μg cm $^{-2}$ thick, which serves both as an electrical contact and a suppressor of the detector's sensitivity to photons.

Electronic pulse height discrimination, together with coincidence logic on the pulses from the two detectors in the telescope, is used to select protons in four energy passbands (nominally 30-80 keV, 80-250 keV, 250-800 keV, and 800-2500 keV) and an integral channel for energies greater than 2.5 MeV. This detector is also sensitive to heavy ions (e.g. He and O) although the particle energies defining the passlands will be marginally higher than those given for protons. A second set of pulse logic isolates events due to ions ($Z>2$) of energies between 6 MeV and 55 MeV.

Table 2.1 lists the nominal energy ranges for the MEPED Detectors. The geometric factor for both the electron and proton directional detector systems is 9.5×10^{-7} m ^{2} sr.

Table 2.1. MEPED detector energy ranges

Data channel	Energy range 90-deg detector and 0-deg detector
Proton Telescope Passbands	
0I and 90I (ions, z >= 2)	6-55 MeV
0P1 and 90P1	30-80 keV
0P2 and 90P2	80-250 keV
0P3 and 90P3	250-800 keV
0P4 and 90P4	800-2,500 keV
0P5 and 90P5	> 2,500 keV
Electron Detector Passbands	
0E1 and 90E1	> 30 keV
0E2 and 90E2	> 100 keV
0E3 and 90E3	> 300 keV

The omnidirectional sensors consist of three nominally identical Kevex Si(Li) solid-state detectors of 50-square-mm area by 3-mm thickness, independently mounted under spherical shell moderators. Each detector has a full-opening viewing angle of 120 degrees in the zenith direction. The omnidirectional flux is defined as the flux through a unit cross-section sphere: Flux = counts/omnidirectional geometric factor.

Table 2.2. MEPED omnidirectional sensors

Data channel	Energy response	Approx. area (solid angle sr)	Omnidirectional geometric factor	Moderator material	thickness
P6	16- 80 MeV	$0.5\ cm^2\ (\pi)$	$1.178\ cm^2\ sr$	Aluminum	0.127 cm
	80-215 MeV	$0.43\ cm^2\ (4\pi)$	$2.701\ cm^2\ sr$		
P7	36- 80 MeV	$0.5\ cm^2\ (\pi)$	$1.178\ cm^2\ sr$	Copper	0.584 cm
	80-215 MeV	$0.43\ cm^2\ (4\pi)$	$2.701\ cm^2\ sr$		
P8	80-215 MeV	$0.43\ cm^2\ (4\pi)$	$2.701\ cm^2\ sr$	Mallory	0.218 cm

The equality of the secondary energy responses of channels P6 and P7 is a reflection of the design decision to equalize the out-of-aperture response of the three omnidirectional sensors. Thus the P8 response can simply be substracted from that of the P6 and P7 channels to obtain their respective primary responses of 16-80 MeV (P6) and 36-80 MeV (P7).

3.0. HEPAD Instrument Description

NOTE: The HEPAD sensor was not been flown on a NOAA spacecraft after NOAA−7. The HEPAD (High Energy Proton and Alpha Detector) senses the intensity in the local zenith direction of ambient solar protons above 370 MeV in four energy bands and of ambient solar alpha particles above 640 MeV/nucleon in two energy bands. Three detectors are employed in a telescope configuration: two solid-state detectors (defining the telescope acceptance aperture) and a Cerenkov radiator/PMT (performing the energy analysis for events producing a triple coincidence in the three detectors).

In-flight calibrations permit the energy-band boundaries to be established to better than $\pm 20\%$:

(1) Large characteristic proton spectrum

$$J(>P) = 2 \times 10^8 \; e^{-P/200} \quad \text{per m}^2 \text{ s sr}$$

$$\text{where: } P = (E^2 + 1876E)^{1/2}$$

(2) Orbit average electron background

$$J(>E) = 8.8 \times 10^9 \; E^{-1} \quad \text{per m}^2 \text{ sec sr}$$

where: E represents particle energy in MeV.

The geometric factor of the telescope acceptance aperture is about 0.9 cm^2 sr with a half-angle field of view of about 24 degrees. Spectral intensity data are supplied at a rate of one sample every 4 seconds. The electronic-circuit logic establishes the detector response as given in Table 3.1.

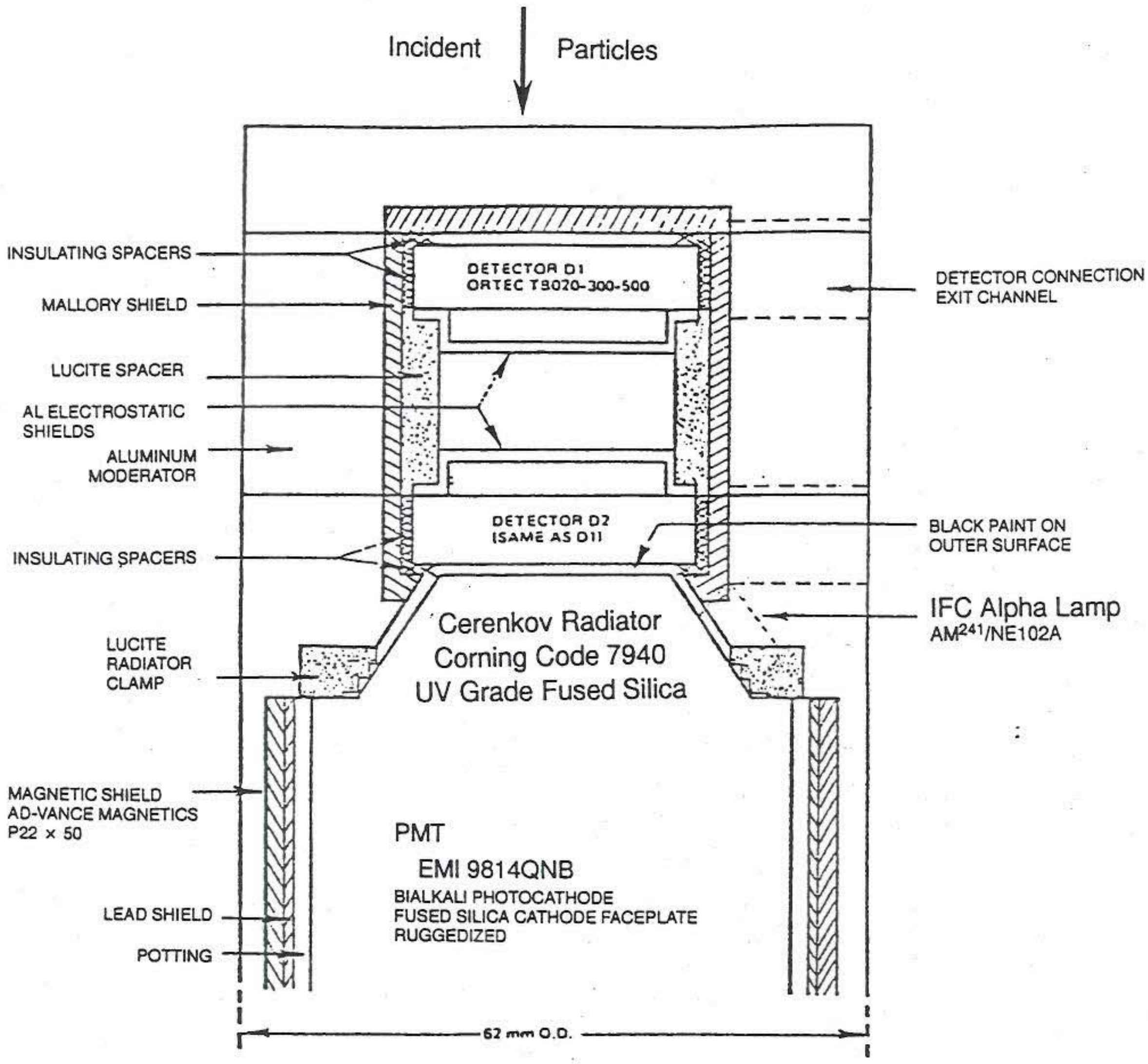
Table 3.1. HEPAD detector outputs

Data channel	Nominal output			Count accumulation interval	Nominal max. random rates (pps)
P1	Protons	370-490	MeV	4 s	620
P2	Protons	490-620	MeV	4 s	420
P3	Protons	620-850	MeV	4 s	260
P4	Protons	>850	MeV	4 s	260
alpha 1	Alphas	600-875	MeV/nucleon	4 s	80
alpha 2	Alphas	>875	MeV/nucleon	4 s	85
S1	1SSD	#1 Singles LS #9		94 ms	1.8×10^5
S2	1SSD	#2 Singles LS #7		94 ms	1.6×10^5
S3	1PMT	Singles LS #1		94 ms	5.6×10^4
S4	1PMT	Gain monitor LS #4		2.5 s	2.0×10^3
S5	1SSD	#1, #2 Double coincidence		1.2 s	2.0×10^4

Data channels S1-S3 identify events exceeding the most sensitive pulse-height-discriminator (LS) thresholds associated with the two SSDs and the PMT; channel S5 identifies time-coincident events in the two SSDs that exceed these thresholds. Channel S4 identifies PMT events, produced by the IFC radioactive source, that exceed the fourth PMT LS threshold.

Figure 3.1 shows a sketch of the telescope assembly. Two surface-barrier silicon detectors D1 and D2 (area 3 cm^2, thickness 500 μm, totally depleted) define an acceptance aperture of about 24 degrees half-angle or geometric factor about $9 \times 10^{-7} m^2$ sr. All linear trajectories passing through these detectors also pass through the conical, fused-silica radiator (special PMT faceplate), which has an average thickness of about 17 mm. For an isotropic environment, the probability distribution of pathlengths in the conical radiator has a mean value 1.05 times the axial thickness of 14 mm, so the average Cerenkov radiation amplitude should correspond to traversal of about 18 mm of silica. Silica is employed as the radiator to provide the desired proton energy threshold (about 320 MeV) and to allow efficient transmission of the shorter wavelengths of the Cerenkov light (cutoff about 190 nm). Most of the area of the radiator's conical surface is bare to allow total internal reflection of incident Cerenkov light from all trajectories within the acceptance cone. Assuming an average quantum efficiency of 18% and full light-collection efficiency within the 200-450 nm interval, 225 photoelectrons should be produced by axial protons of $\beta = 1$.

Mallory metal (high-Z) is employed to shield the detectors against bremsstrahlung generated by ambient electrons (its thickness is one absorption length for E<350 keV). Similarly, an aluminum moderator (low-Z) is employed to shield these detectors against ambient electrons and protons, and to suppress the bremsstrahlung radiated by the stopping electrons. Within the out-of-aperture solid angle, the moderator will stop protons of <80 MeV and electrons of <7 MeV. For in-aperture directions, the shielding is effective against protons of <65 MeV and electrons of <4 MeV and will absorb about 15 MeV from a 370-MeV proton. Shielding of the detectors from "upward-"entering protons of E<90 MeV is supplied by the silica radiator and the magnetic shield, lead shield, and aluminum shell surrounding the PMT. Beginning with NOAA-8 the HEPAD instrument has no longer been included as part of the SEM although the archive data record continued to include those data channels which, after NOAA-7, contained default numbers.



DETECTOR AREA: 3 cm^2
DETECTOR SEPARATION: 29 cm
GEOMETRIC FACTOR: G = 0.9 cm^2 sr
ACCEPTANCE APERTURE: 68% of G WITHIN <24° HALF-ANGLE
100% of G WITHIN <34° HALF-ANGLE

Figure 3.1. HEPAD telescope assembly.

4.0. TED Instrument Description

The TED (Total Energy Detector) measures the total energy flux carried into the atmosphere by charged particles of auroral energies.

Four separate electrostatic-analyzer, charged-particle detector systems were included on TIROS/NOAA and were mounted in pairs. Within each pair, one analyzer system is devoted to measuring electrons and the other to measuring positive ions (protons). The two pairs view charged particles coming from different directions so that observations can be made of the directional energy flux at two different angles to the local geomagnetic field direction. One of these pairs views outward, parallel to the Earth center radial vector, so that it measures charged particles whose velocities are toward Earth along this radial vector. The other pair views at an angle of 30 degrees to the first. In this documentation, data from the first pair are tagged with the prefix 0 and data from the second pair with the prefix 30. It is stressed that these two angles are defined with respect to the Earth center-satellite vector and have nothing to do with the pitch angle α associated with the charged particles being measured.

The two detector systems within each pair are alternated between measuring electrons and measuring protons. The time taken for a full cycle is 2 s. The first half cycle (1 s) is devoted to measuring electrons. This 1-s period is divided into 13 equal segments. During the first 1/13 s, a background measurement is taken; during the final 1/13 s, the instrument undergoes a reset sequence during which no data are taken. During the center 11/13 s, the analyzers are swept, effectively linearly with time, from an energy of 300 eV to an energy of 20,000 eV. The total number of counts accumulated by the detector during this sweep is telemetered to the ground as a measure of the integrated (from 300 eV to 20,000 eV) directional energy flux carried by the electrons observed by that particular detector. During the second half of the cycle the process is repeated for protons using the second detector system.

4.1. Terminology and Transmission Sequence

0EF-D and 30EF-D: The total number of counts accumulated during a sweep on the 0- and 30-degree electron detectors. These numbers are, to first approximation, linearly related to the directional energy flux carried by electrons between 300 eV and 20,000 eV.

0PF-D and 30PF-D: The corresponding total accumulated counts for the 0- and 30-degree proton detectors during the second part of the 2 s instrument cycle.

The four data points (0EF-D, 30EF-D, 0PF-D, and 30PF-D) are regarded as the prime data from the TED instrument and are transmitted continuously every 2 s (about 15 km of spacecraft travel).

The TIROS/NOAA instrument also telemeters data points that are related to the directional energy fluxes associated with electrons or ions having energies within selected, narrow energy bands. Thus, an estimate can be made of which energy particles are carrying the bulk of the energy flow and at what altitude in the atmosphere this energy will ultimately be deposited.

The maximum count accumulated in a single 1/13-s subinterval during a given sweep of a given detector system and the corresponding energy band number are transmitted every 2 s from both detector pairs for both particle species. These data represent both the energy band containing the most energy flux and the value of that directional energy flux:

0DE-M, 30DE-M, 0DP-M, 30DP-M: The maximum count

0E-M, 30E-M, 0P-M, 30P-M: The corresponding interval number.

During the full energy sweep from 300 eV to 20,000 eV, which takes 11/13 s, the number of detector counts are accumulated during each successive 1/13 s interval, potentially giving 11 data points from each detector during an energy sweep. Since there are two detector pairs studying two particle species, there are potentially four sets of measurements, each containing 11 energy channels generated every 2 s. However, data concerning only one of these energy channels is transmitted on each detector sweep. This channel is the one that has accumulated the greatest number of counts during the sweep. The channel number and the accumulated counts are transmitted from each detector once every 2-s experiment cycle. In addition, and at a lower duty cycle, the counts accumulated by each detector within energy channels 1, 3, 5, and 7 are transmitted to provide four-point energy spectra. This lower-duty cycle lasts 8 s and involves transmitting in sequence the four-point measurements from the 0-degree electron, 30-degree electron, 0-degree proton, and 30-degree proton detector systems. The latter cycle is interrupted every fourth 8-s cycle so that background count-rate data from each detector can be transmitted.

Background data are also included as a quality check on the operation of the instrument. The counts registered by each detector during the first 1/13 s (background phase) of each sweep are accumulated for 16 sweeps—a total of 1.23 s. The accumulated counts 0E-BK, 0P-BK, 30E-BK, and 30P-BK are transmitted once each 32 s in place of the normal transmission of 0DE-1, 0DE-3, 0DE-5, and 0DE-7. These counts generally number less than 50; if they exceed 200, possible instrument malfunctions ought to be considered.

4.2. Conversion

There are two physical interpretations of the counts accumulated during a single subinterval of the energy sweep. The first relates this number to the directional energy flux within the limited energy range swept by the detector in the 1/13-s subinterval. The second is obtained by dividing the directional energy flux by the width of the energy band sampled, thus obtaining the directional differential energy flux at the center energy of the band.

The conversion between the telemetered values and the corresponding directional energy flux in physical units is shown below. The difference between electron and proton conversion reflects a difference in the detection efficiencies of the two particle species:

Directional energy flux mW/($m^2 \cdot sr$) =

For electrons: 1.905×10^{-3} × (0EF-D or 30EF-D)
For protons: 1.50×10^{-3} × (0PF-D or 30PF-D)

A data point can be multiplied by the following conversion factors to convert from counts to differential directional energy flux:

Differential directional energy flux mW/($m^2 \cdot sr \cdot eV$) at the center energy =

For electrons 3.78×10^{-7} (for all energy bands)
For protons 2.97×10^{-7} (for all energy bands)

Table 4.2 lists the details of each of the energy bands and gives the multiplying constants required to convert the raw data point to a physical quantity; it also shows the altitudes in the atmosphere where electrons within each band will deposit their energy.

Table 4.2. TED energy bands and altitudes of energy deposition

Energy band number	Edges of band (eV)	Center energy (eV)	Conversion from counts to directional energy flux mW/($m^2 \cdot$ sr)		Altitude at which energy will be deposited (km)
			Electrons	Protons	
1	300–458	379	5.97×10^{-5}	4.69×10^{-5}	>300
2	458–773	616	1.19×10^{-4}	9.38×10^{-5}	215
3	773–1088	931	1.19×10^{-4}	9.38×10^{-5}	190
4	1088–1718	1403	2.38×10^{-4}	1.88×10^{-4}	165
5	1718–2349	2033	2.38×10^{-4}	1.88×10^{-4}	145
6	2349–3610	2979	4.76×10^{-4}	3.75×10^{-4}	130
7	3610–4870	4250	4.76×10^{-4}	3.75×10^{-4}	120
8	4870–7392	6131	9.52×10^{-4}	7.50×10^{-4}	115
9	7392–9914	8653	9.52×10^{-4}	7.50×10^{-4}	108
10	9914–14957	12436	1.90×10^{-3}	1.50×10^{-3}	105
11	14957–20000	17479	1.90×10^{-3}	1.50×10^{-3}	104

The directional energy fluxes obtained using the conversion factors in Table 4.2 are appropriate to the satellite location at 850km. The measured values must be manipulated, together with a geomagnetic field model, to obtain the truly relevant parameter: the magnitude of the energy flow into the atmosphere and the location at which this energy input is occurring.

Figure 4.1 illustrates this situation. The charged particles measured at the satellite are guided along the magnetic field lines. Because these lines of force are not radial, the point at which the field line that passes through the satellite actually intersects the atmosphere may be displaced considerably from the subsatellite point. In the TIROS/NOAA data-processing system, a magnetic field model is used to trace the field line passing through the satellite to the point where the field line intersects the atmosphere at 120 km (Foot Of the Field Line, FOFL). The coordinates of this point (both geographic and geomagnetic), together with the solar time and magnetic time, are calculated and given in the archive tape record header data. By convention, if TIROS/NOAA is north of the geomagnetic equator, the FOFL is taken to be in the Northern Hemisphere; otherwise it is in the Southern Hemisphere.

The angles between the geomagnetic field direction and the look direction of the two detector pairs are also computed using the same geomagnetic field model. The supplement of these two angles is the local pitch angles of the charged particles being studied by the two detector pairs. However, because of the "magnetic mirror effect" on the motion of charged particles, these particles do not have the same pitch angles at the location of the satellite as they would have at the top of the atmosphere (120 km). The relation between the two pitch angles is

$$\sin \alpha_{120} = \sqrt{\frac{B_{120}}{B_{850}}} \sin \alpha_{850}$$

where α_{120} = particle pitch angle at the FOFL,

α_{850} = particle pitch angle at the TIROS/NOAA spacecraft,

B_{120} = geomagnetic field strength at the FOFL, and

B_{850} = geomagnetic field strength at the TIROS/NOAA spacecraft.

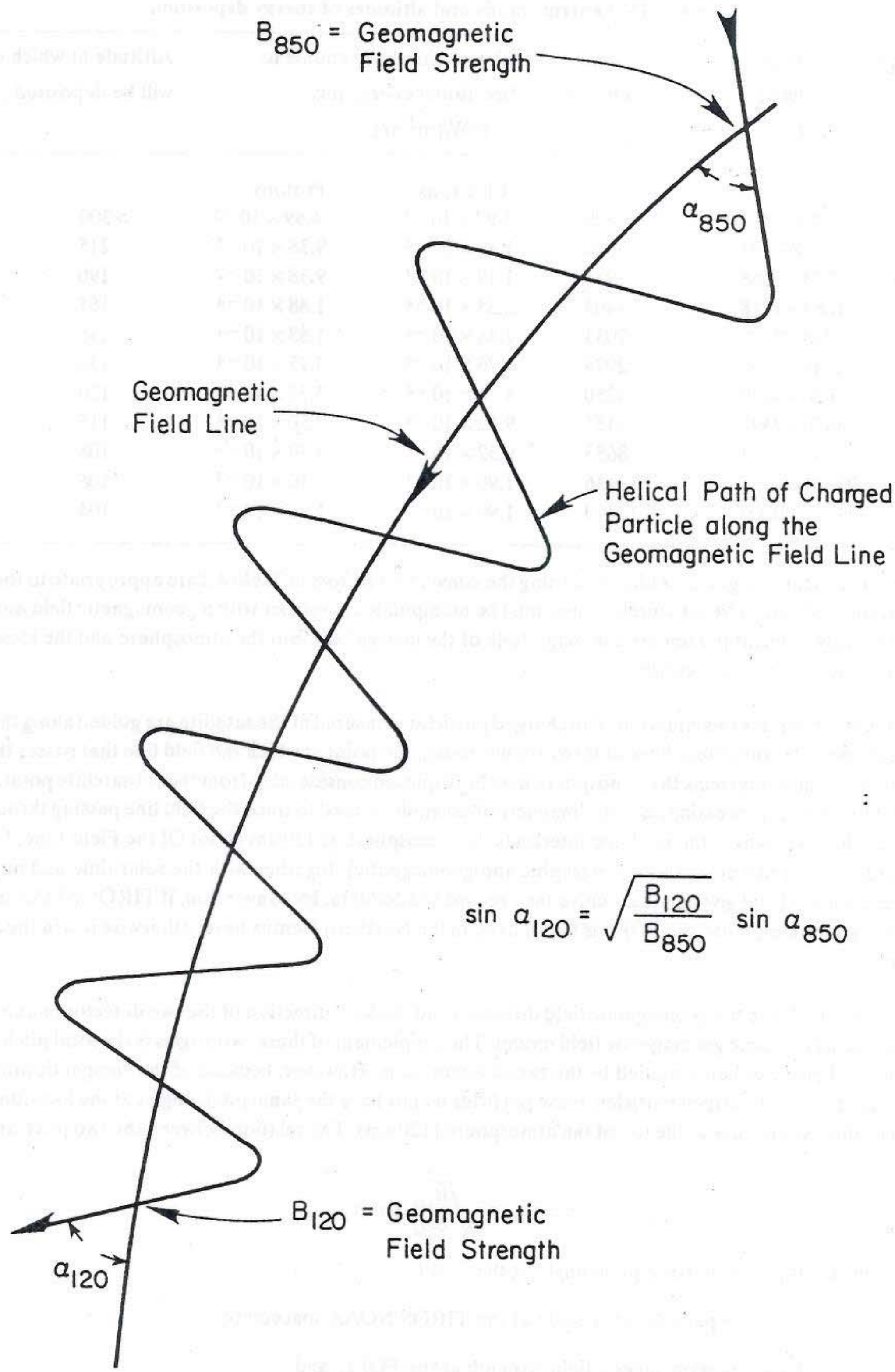


Figure 4.1. Path of charged particles along the geomagnetic field line.

Figure 4.1 illustrates how the pitch angle of a charged particle varies as it moves along the geomagnetic field line between TIROS/NOAA and the atmosphere. Note that convention defines a particle's velocity vector and the direction of the magnetic field. This means that, in the Northern Hemisphere, charged particles moving downward toward the atmosphere have pitch angles between 0 and 90 degrees. In the Southern Hemisphere, charged particles moving toward the atmosphere have pitch angles between 90 and 180 degrees.

Note also that it is possible for the sin α_{120} to exceed 1.0. Physically, this occurs when the charged particles measured as they move downward toward the atmosphere at TIROS/NOAA in fact magnetically mirror before reaching the atmosphere and return back up the magnetic field line. Such particles cannot be counted as contributing to the energy influx into Earth's atmosphere. In the course of data processing, all parameters concerning the geomagnetic field are computed once each 8 s and given in the archive record's header data. These parameters are included:

(1) The three-vector components of the geomagnetic field at TIROS/NOAA and the scalar magnitude of the field.

(2) The geographic location where the geomagnetic flux tube that threads TIROS/NOAA intersects the top of the atmosphere at 120 km.

(3) The geomagnetic coordinates of the FOFL and the local solar and geomagnetic times of the FOFL.

(4) The three-vector components of the geomagnetic field at the FOFL and the scalar magnitude of the field.

(5) The pitch angles of those charged particles being observed by the two TED detector systems as transformed to the FOFL.

By using the measurements of 0EF-D, 30EF-D, 0PF-D, and 30PF-D, together with the pitch angles at which the measurements were made (as transformed to 120 km), the total energy flux

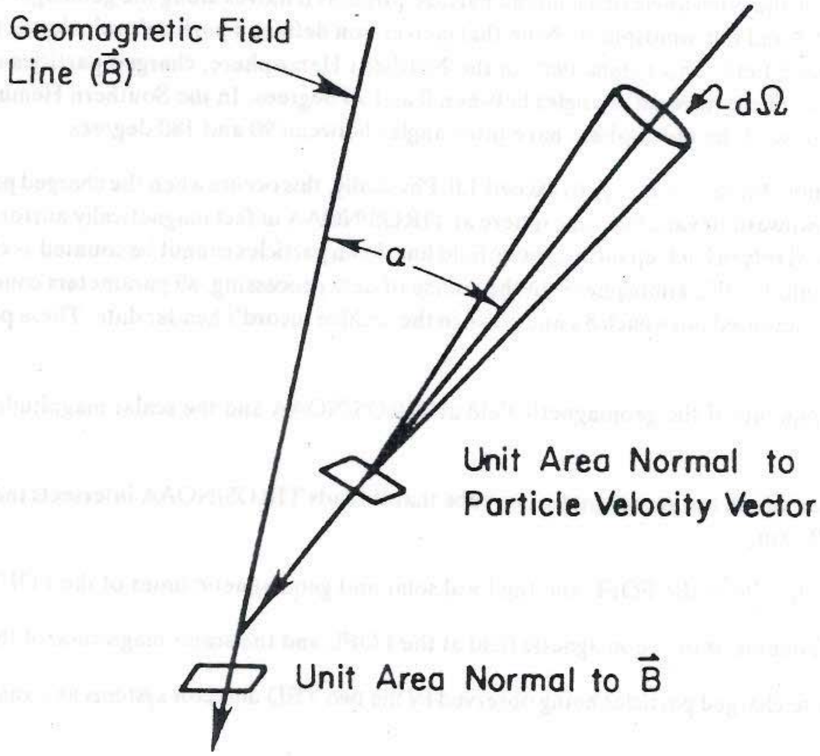
$$E_T = 2\pi \int_0^{\pi/2} E_D(a) \sin a \cos a \; d\, a$$

may be evaluated independently for both electrons and ions. The geometry defining the integration of directional energy fluxes over angle is illustrated in Figure 4.2. The sum of these two is converted to physical units ($\mu W/m^2$) and becomes the "Total Energy Flux" (TEDFX) value given in the TED format.

If neither of the two detector pairs is viewing charged particles that reach Earth's atmosphere, then the value of the Total Energy is set to 0. However, the actual values of 0EF-D, 30EF-D, 0PF-D, and 30PF-D remain available in the output record. The situation in which neither detector pair views charged particles that can reach the atmosphere is confined to measurements made at rather low geographic latitudes, where any energy flow into the atmosphere is expected to be small.

The TED detector system was originally designed to be operated in any one of three different modes, to compensate for possible detector failures. However, because the instrument has thus far proved reliable, there is a policy that the TED instrument is operated only in its normal mode.

The reprocessing of historical TIROS/NOAA SEM observations that was done in 1993 allowed much-improved data-quality control to be introduced into the software. In particular, whenever bad data were identified within a 2-second data record, the corresponding entry for the total energy flux was set to 1000 (an energy flux beyond the maximum that could be detected by the instrument). Data records that are associated with an energy flux of less than 1 W/m^2 may be regarded as valid.



$j(E,\alpha)$ = Particle Differential Directional Number Flux

$E(E,\alpha) = j(E,\alpha)E$ = Particle Differential Direction Energy Flux

$$E_D(\alpha) = \int_0^{\infty} j(E,\alpha)\,E\,dE$$ = Particle Directional Energy Flux

$$E_T = 2\pi \int_0^{\pi/2} E_D(\alpha)\sin\alpha\cos\alpha\,d\alpha$$ = Particle Total Energy Flux (Evaluated at top of atmosphere)

Figure 4.2. Evaluation of particle total energy flux.

5.0. Data Descriptions

This section specifies the data contained on the archive tapes and gives the timing information necessary to convert the data to usable units. In general the data can be divided into five types of information: orbital information, housekeeping information, MEPED data, HEPAD data, and TED data.

Orbital information contains the time of the data record and defines the location and orientation of the spacecraft at that time.

Housekeeping information specifies the status of each individual instrument (*on* or *off*) and whether an in-flight calibration is in progress. Weekly calibrations of the SEM instruments appear on the archive tapes, and it is the user's responsibility to recognize and discard calibration data.

The MEPED, HEPAD, and TED data descriptions identify the data channels, detectors, particle types, and energy ranges on each instrument. The descriptions also include the channel name abbreviations used throughout this document and the record-timing tables.

The archive tapes consist of short records, each containing 8 seconds of data. The begin time, called T0, is included on every record. T0 is the time of the orbital and housekeeping information on that record. However, T0 is not the time at which the MEPED, HEPAD, and TED data channels were sampled on the spacecraft. The record-timing tables (in Sections 5.3, 5.4, and 5.5) show the time, relative to TO, when each data channel began accumulation. The length of the accumulation period is also included so that counts per accumulation period can be converted into counts per second. An example of the record-timing table for the MEPED data is shown below. In this example the MEPED data channels 0E1 to 0E3 are accumulated for 1 s each. The first readings of each of these data channels (MEPED words 6, 7 and 8) represent count accumulation that began at time T0 -1 second. The second readings of each data channel (MEPED words 25, 26 and 27) began accumulation at T0 +1 s. The user can determine the exact time a data channel was sampled and convert the value to "counts per second" by referring to the record-timing tables.

Table 5.1. Sample record-timing table

Word	Data channel	Accumulation period(s)	Accumulation begin time relative to T0(s)
.	.	.	.
.	.	.	.
6-8	0E1 to 0E3	1	-1
25-27	0E1 to 0E3	1	+1

5.1. Orbital and Spacecraft Information

Each data record contains orbital and spacecraft information plus date and time of the record (T0), the location and orientation of the spacecraft, and other values useful in evaluating the instrument data. Orbital information is valid for T0. Latitudes, longitudes, pitch angles, local times, and magnetic times are given in degrees. Magnetic field parameters are in nanotesla. Orbital information includes original orbital data and calculated values (e.g. pitch angles).

The calculated values are helpful in interpreting the instrument data. Magnetic field and FOFL values, necessary to interpret the charged particle data, are calculated from the satellite location using a magnetic field model. The model uses a look-up table and an interpolation scheme. The geomagnetic field parameters look-up table (Aarnes and Lundblad, University of Bergen, Norway; personal communication, 1979.) is a 4-degree-latitude by 4-degree-longitude grid of subsatellite locations. Magnetic values for this table were computed using the model magnetic field and field line tracing routines developed at the National Space Science Data Center (Stassinopoulos, E. G., and G .D. Mead, 1972. ALLMAG, GDALNG, LINTRA: Computer programs for geomagnetic field and field-line calculations. NASA/NSSDC Report 72-12.). The table was constructed for a single satellite altitude of 870 km, and, strictly speaking, the table and interpolations are appropriate only for a satellite at that altitude. The TIROS/NOAA satellite are at altitudes ranging from 815 to 850 km, so errors are introduced by the procedure. However, at high latitudes, where the charged particle observations are most useful, these errors are not large.

The satellite is moving in a $-Y$ direction.

Following are brief descriptions of the orbital and spacecraft information. Sections 6.1. and 6.5. contain the orbital and spacecraft data formats, ranges of values, units, and conversion factors.

Spacecraft ID

0 = zero fill record
1 = TIROS-N
2 = NOAA-6
4 = NOAA-7
6 = NOAA-8
8 = NOAA-10
5 = NOAA-12
3 = NOAA-14

T0, begin date/time of the record

The begin date/time of a record is referred to as T0 and is defined by the year, day of the year, and milliseconds of the day in Universal Time. Each record contains 8 s of data and records are ordered by time. There are no duplicate records, and the time difference between records is always a multiple of 8 s. All orbital and housekeeping information is valid at T0.

Note for users decoding the records: the record format separates the year (bytes 141, 142) and day of year (bytes 143,144) from the milliseconds of the day (bytes 1,2,3,4). Hovever, the supplied application routines group them back together in words IHD(2), IHD(3), and IHD(4).

Receiving station

The station that recorded the original raw data is available on the archive tape. Table 6.5. shows where the receiving station is stored. The receiving station does not affect the archive data in any way.

Station Codes:
0 = unknown
1 = Wallops Island, Virginia
2 = Gillmore Creek, Alaska
3 = Western European Stations, Spain

Altitude of the satellite

TIROS/NOAA satellites are in near-circular orbits of about 850 km. In calculations for the archive tape involving altitude, 870 km is used for all spacecraft. There is no record of the actual altitude for each spacecraft but the nominal altitude for each spacecraft is included on the archive tapes for the user's information, see Table 6.5.

TIROS-N = 853.5 km
NOAA-6 = 815.5 km
NOAA-7 = 849.2 km
NOAA-8 = 815.5 km
NOAA-10 = 833.0 km
NOAA-12 = 815.0 km
NOAA-13 = 870.0 km
NOAA-14 = 870.0 km

Inclination of the satellite

The satellite inclination is also assumed to be constant. The nominal inclinations are written on the archive tapes for each spacecraft, see Table 6.5.

TIROS-N = 98.9°
NOAA-6 = 98.7°
NOAA-7 = 98.9°
NOAA-8 = 98.9°
NOAA-10 = 98.7°
NOAA-12 = 98.7°
NOAA-13 = 98.9°
NOAA-14 = 98.8°

Orbit number

The orbit number is a count of the number of times the satellite has orbited Earth. The orbit number, or revolution number, is incremented when the satellite crosses the Equator from the Southern to the Northern Hemisphere. The orbital period of all spacecraft is approximately 102 minutes. Orbit numbers are based on time and estimated orbital period. They are updated only in record type 1. See Table 6.5.

Record type

On the original raw data tapes each logical record contains 32 s of data. To simplify the archive data records these 32-s records are divided into four 8-s records. The four 8-s records all contain the same information except for slight differences in MEPED and TED data. These differences are described in Sections 6.6. and 6.8. To flag these differences, each 8-s record contains a "record type," a number from 1 to 4 that denotes the record's location in the original raw data record, see Table 6.5.

Data at the Satellite

Geocentric latitude at the satellite

Geographic east longitude at the satellite

The geocentric (geographic) latitude and geographic east longitude are the subsatellite location. They are valid for time T0. See Table 6.2.

BR: the radial component of the magnetic field strength in nT at the satellite, where positive is up.

BT: the north-south component at the satellite, where positive is south.

BP: the east-west component at the satellite, where positive is east.

BB: the total field at the satellite.

This information has been calculated using the latitude and longitude at the satellite. See Table 6.2.

Data at the Foot of the Field Line (FOFL)

The FOFL is the point where the magnetic field line through the satellite crosses 120 km altitude. The model used to calculate the magnetic field vector at a given location is also used to trace the magnetic field line through the satellite to an altitude of 120 km in the local hemisphere.

Geographic latitude at the FOFL

Geographic east longitude at the FOFL

The geographic latitude and geographic east longitude at the FOFL are calculated from the latitude and longitude at the satellite and are valid for time T0. See Table 6.2.

BR120: the radial component of the magnetic field strength in nT at the FOFL, where positive is up.

BT120: the north-south component at the FOFL, where positive is south.

BP120: the east-west component at the FOFL, where positive east.

BB120: the total field at the FOFL.

This information has been calculated using the FOFL location. See Table 6.2.

Geomagnetic latitude at the FOFL

Geomagnetic east longitude at the FOFL

The geographic latitude and longitude at the FOFL are used as the input to a simple transformation to obtain these *uncorrected* dipole geomagnetic coordinates, see Table 6.2.

L-value at the FOFL

The L value is also computed, in the model, at the FOFL. If the L value is computed to be ≥15.00 it is considered to be undefined and the value is set to 0.00. This causes no ambiguity because even at the magnetic equator the L value sampled by the satellite is never less than about 1.10. The L-values computed from the magnetic field model have not been thoroughly verified and should be used with caution. The orbital information is sufficient, however, to permit the user to recompute the L-value from updated models. See Table 6.2.

Pitch Angles

The pitch angles, with respect to the geomagnetic field, of the particles being sensed by the TED and MEPED detectors are calculated to help the users interpret the data. The pitch angle of the charged particles being sampled by a detector is the angle between the particle's velocity vector and the magnetic field. It is defined with the convention that 0 deg is a particle moving parallel to the magnetic field and 180 deg is a particle moving antiparallel to the magnetic field. The look direction of the detectors and three components of the magnetic field, together with the location of the satellite, are used to compute the pitch angles of the particles being measured at the satellite.

The calculations for the TED and MEPED detector systems are performed as follows:

TED0: The pitch angle of the particle being sensed by the TED 0-degree detector as transformed to the FOFL.

TED30: The pitch angle of the particle being sensed by the TED 30-degree detector as transformed to the FOFL.

These pitch angles have been transformed to the FOFL. If the particle mirrors above 120 km altitude, the pitch angle is given as 90 degrees The pitch angles assigned to the TED detectors (which view along the Earth-center radial vector and at 30 degrees to that direction) are the pitch angles possessed by the particles and have been transformed from the satellite location to the FOFL to ease the integration over angle required to complete the energy flux *into* the atmosphere. See Table 6.2.

MEPED81: MEPED 90-degree proton-detector pitch angle at the satellite.

MEPED83: MEPED 90-degree electron-detector pitch angle at the satellite.

MEPED0: MEPED 0-degree proton- and electron-detector pitch angle at the satellite.

In the original version of the archive data, the pitch angles assigned to the MEPED detectors were miscalculated whenever the satellite was southbound. This error has been corrected in the reprocessing of historical data for reformatting onto 3480 tape cartridges. Moreover, the same pitch-angle convention has been adopted for both the TED and MEPED particle observations. That is, the pitch angle that is given is the angle between the particle's velocity vector and the magnetic field vector, so particles that precipitate directly into the atmosphere in the northern hemisphere have pitch angles near 0 degrees and in the southern hemisphere near 180 degrees. See Table 6.2.

Miscellaneous Information

Local time and *Magnetic local time*

The approximate local solar time at the subsatellite location is computed from the universal time T0 and the geographic longitude. Similarly, the eccentric dipole magnetic local time *at the subsatellite location* is computed from the subsatellite location, day, and universal time.

The local and magnetic local times are in degrees eastward from midnight and can be converted to hours by dividing by 15. The local time at the FOFL can be calculated as follows,where TIME is the local time in hours, and TIME multiplied by 15 is the local time in degrees.

MIN = milliseconds of the day/3,600,000,

TIME = MIN + (FOFL geographic longitude in degrees/15).

If TIME > 24 then TIME = TIME - 24,

Program version

The Program Version number will be incremented whenever a change is made in the archive tape data or format. It is currently 2.

5.2. Housekeeping and Status Information

The housekeeping and status information on the archive tapes include instrument on/off flags, in-flight calibration (IFC) flags, TED flags, and various housekeeping values. All the flags and values are valid at T0, the begin time of the record. See Tables 6.3. and 6.5.

Instrument flags

The instrument on/off flags are normally in the "on" state but should always be checked. On occasion, an instrument has been mysteriously commanded "off" and remained so for up to 2 weeks.

IFC flags

In-flight calibrations (IFCs) are conducted approximately once per week, and they appear on the tapes in the same format as the other data. If the IFC flag is equal to 1 (yes), an IFC is in progress. The IFC flags should be checked on every record, and the data should be discarded when an IFC is in progress. The calibration sequence begins for all instruments at the same time, but the MEPED IFC lasts for 9 minutes 36 seconds and the TED/HEPAD lasts for 12 minutes 48 seconds. Sometimes, however, only partial calibrations are on the tape. This happens when the flags are set by telemetry noise and/or when data are missing from the archive tape because of telemetry problems.

TED flags

The TED MODE and TELEMETRY FORMAT flags together show what data are being sampled in the TED experiment. The following configurations are possible, but only the first setting (MODE 0 or 2, FORMAT 1) has been or is expected to be used.

TED MODE and TELEMETRY FORMAT descriptions:

MODE 0 or 2, FORMAT 1—normal electrons and protons

MODE 1, FORMAT 1—electron dwell, no protons

MODE 3, FORMAT 1—proton dwell, no electrons.

For 104 minutes after an IFC, the TED channeltron gain is verified by cycling the setting of the pulse height discriminators. This usually does not affect the data, but the user can check for this condition from the TED pulse height discrimator (PHD) flags. The normal value of the PHD flags is all zeros; other values, particularly varying values, ordinarily mean the TED channeltron gain is being verified.

Housekeeping Values

No explanation of the housekeeping values is included here because they are solely for SEL's use in evaluating the performance of the instruments. Table 5.2 is a list of the housekeeping word abbreviations and their meanings.

Table 5.2. Housekeeping word descriptions

Detector Element			Description
1	MPTT	MEPED	proton telescope temperature
2	METT	MEPED	electron telescope temperature
3	MELT	MEPED	electronics temperature
4	OMNI	MEPED	OMNI temperature
5	AMSS	MEPED	detector-bias voltage
6	HELT	HEPAD	electronics temperature
7	PMTT	HEPAD	photo-multiplier tube temperature
8	PMHV	HEPAD	high-voltage power-supply monitor
9	HSSD	HEPAD	solid-state detector-bias monitor
10	LVL	HEPAD	level calculated from PMHV and HELT
11	TEPS	TED	electron channeltron power-supply monitor
12	TPPS	TED	proton channeltron power-supply monitor
13	LVR	TED	low-voltage ramp monitor
14	CEA	TED	cylindrical electrostatic analyzer power-supply monitor
15	TEDT	TED	temperature monitor

5.3. MEPED Data Description

MEPED data channels are directional measurements, at 0 and approximately 80 degrees (with respect to the local zenith) for electron channels E1 to E5, proton channels P1 to P5, and the ion ($z \geq 2$) channels. Channels P6, P7, and P8 provide omni-directional measurements. The geometric factor for both the proton telescope and electron telescope is $9.5 \times 10^{-3} cm^2$ sr.

Table 5.3. MEPED data channels

MEPED data Element	channel	Nominal energy Detector	Particle type	range keV
1	0I	0° Ions	Pos. (Z ≥2) Ions	6,000–55,000
2	90I	90° Ions	Pos. (Z ≥2) Ions	6,000–55,000
3	0P1	0° Proton	Protons	30–80
4	0P2	0° Proton	Protons	80–250
5	0P3	0° Proton	Protons	250–800
6	0P4	0° Proton	Protons	800–2,500
7	0P5	0° Proton	Protons	>2,500
8	0E1	0° Electron	Electrons	>30
9	0E2	0° Electron	Electrons	>100
10	0E3	0° Electron	Electrons	>300
11	90P1	90° Proton	Protons	30–80
12	90P2	90° Proton	Protons	80–250
13	90P3	90° Proton	Protons	250–800
14	90P4	90° Proton	Protons	800–2,500
15	90P5	90° Proton	Protons	>2,500
16	90E1	90° Electron	Electrons	>30
17	90E2	90° Electron	Electrons	>100
18	90E3	90° Electron	Electrons	>300
19	P6	Omnidirectional Proton	Protons	16–80 MeV 80–215 MeV
20	P7	Omnidirectional Proton	Protons	36–80 MeV 80–215 MeV
21	P8	Omnidirectional Proton	Protons	80–215 MeV

On the MEPED, each data channel is sampled four times every 8 s (except for the 0 I and 90 I channels, which are sampled only once every 16 s). The record-timing information for the MEPED is shown in Table 5.4.

Table 5.4. MEPED data—record timing

MEPED Element	Data channel	Accumulation period (s)	Accumulation begin time relative to T0 (s)
1	0 I	16	-16
2	90 I	16	-16
3-7	0P1 to 0P5	1	-1
8-10	0E1 to 0E3	1	-1
11-15	90P1 to 90P5	1	+0
16-18	90E1 to 90E3	1	+0
19-21	P6 to P8	2	-2
22-26	0P1 to 0P5	1	+1
27-29	0E1 to 0E3	1	+1
30-34	90P1 to 90P5	1	+2
35-37	90E1 to 90E3	1	+2
38-40	P6 to P8	2	+0
41-45	0P1 to 0P5	1	+3
46-48	0E1 to 0E3	1	+3
49-53	90P1 to 90P5	1	+4
54-56	90E1 to 90E3	1	+4
57-59	P6 to P8	2	+2
60-64	0P1 to 0P5	1	+5
65-67	0E1 to 0E3	1	+5
68-72	90P1 to 90P5	1	+6
73-75	90E1 to 90E3	1	+6
76-78	P6 to P8	2	+4

5.4. HEPAD Data Description

The HEPAD data channels are shown in Table 5.5. Section 3 describes the instrument. No HEPAD sensor after NOAA-7.

Table 5.5. HEPAD data channels

HEPAD Element	Data channel	Detector	Particle type	Energy range
1	P1	Proton	Protons	370–480 MeV
2	P2	Proton	Protons	480–640 MeV
3	P3	Proton	Protons	640–850 MeV
4	P4	Proton	Protons	>850 MeV
5	alpha 1		Alphas	640–850 MeV/Nucleon
6	alpha 2		Alphas	>850 MeV/Nucleon
7	S5			SSD D1-D2 Double Coincidences
8	S4			PMT Gain Monitor
9	S1			SSD D1 Singles
10	S2			SSD D2 Singles
11	S3			LSI (PMT Anode) Singles

The HEPAD data channels are read out every 4 s; therefore, there are two values for each channel in every 8-s data record. The record-timing information for the HEPED is shown in Table 5.6.

Table 5.6. HEPAD data–record timing

HEPAD Element	Data channel	Accumulation period (s)	Accumulation begin time relative to T0 (s)	HEPAD Element	Data channel	Accumulation period (s)	Accumulation begin time relative to T0 (s)
1	P1	4.0	-4.0	12	P1	4.0	+0.0
2	P2	4.0	-4.0	13	P2	4.0	+0.0
3	P3	4.0	-4.0	14	P3	4.0	+0.0
4	P4	4.0	-4.0	15	P4	4.0	+0.0
5	alpha 1	4.0	-4.0	16	alpha 1	4.0	+0.0
6	alpha 2	4.0	-4.0	17	alpha 2	4.0	+0.0
7	S5	1.2	-1.2	18	S5	1.2	+3.2
8	S4	2.5	+0.0	19	S4	2.5	+4.0
9	S1	0.1	+2.5	20	S1	0.1	+6.5
10	S2	0.1	+2.6	21	S2	0.1	+6.6
11	S3	0.1	+2.7	22	S3	0.1	+6.7

5.5. TED Data Description

The TED instrument and its data channels are described in Section 4. The TED data in each 8-s data record contains values from four different detectors, each sampled once every 2 s. Table 5.7 describes the data channels for each detector. Table 5.8 shows the sequencing of the data values, as they appear on the archive tape, and the record-timing information.

Table 5.7. TED data channels

Data channel	Detector	Parameters	Energy interval during which counts accumulated
0DE-1	0° electron	differential directional energy flux	1st interval
0DE-3	0° electron	"	3rd interval
0DE-5	0° electron	"	5th interval
0DE-7	0° electron	"	7th interval
0DE-M	0° electron	"	maximum interval
0E-M		interval	interval max counts accumulated
0EF-D	0° electron	directional energy flux	all intervals
30DE-1	30° electron	differential directional energy flux	1st interval
30DE-3	30° electron	"	3rd interval
30DE-5	30° electron	"	5th interval
30DE-7	30° electron	"	7th interval
30DE-M	30° electron	"	maximum interval
30E-M		interval	interval max counts accumulated
30EF-D	30° electron	directional energy flux	all intervals
0DP-1	0° proton	differential directional energy flux	1st interval
0DP-3	0° proton	"	3rd interval
0DP-5	0° proton	"	5th interval
0DP-7	0° proton	"	7th interval
0DP-M	0° proton	"	maximum interval
0P- M		interval	interval max counts accumulated
0PF-D	0° proton	directional energy flux	all intervals
30DP-1	30° proton	differential directional energy flux	1st interval
30DP-3	30° proton	"	3rd interval
30DP-5	30° proton	"	5th interval
30DP-7	30° proton	"	7th interval
30DP-M	30° proton	"	maximum interval
30P M		interval	interval max counts accumulated
30PF-D	30° proton	directional energy flux	all intervals
Total Energy	all detectors	total energy flux	all intervals

Table 5.8 shows the TED record-timing information for the TED in normal mode, telemetry formats 0 and 2. This is the only configuration in which the TED has been operated. The accumulation period, written as 11/13, means eleven-thirteenths of a second. Note that the TED record contains four groups of data. The first 6 words of each group are from a different detector, but the remaining 14 words are the same detectors read every 2 s.

Table 5.8. TED data—record timing, for record types 1, 2, & 3

TED Element	Data channel	Accumulation period (s)	Accumulation begin time relative to T0 (s)
1-4	0DE-1, 0DE-3, 0DE-5, 0DE-7	11/13	-2
5	0DE-M	11/13	-2
6	0E-M	-	-
7-8	0EF-D, 0DE-M	11/13	-2
9	0E-M	-	-
10-11	30EF-D,30DE-M	11/13	-2
12	30E-M		-
13-14	0PF-D, 0DP-M	11/13	-1
15	0P-M	-	-
16-17	30PF-D,30DP-M	11/13	-1
18	30P-M	-	-
19-22	30DE-1, 30DE-3, 30DE-5, 30DE-7	11/13	+0
23	30DE-M	11/13	+0
24	30E-M	-	-
25-26	0EF-D, 0DE-M	11/13	+0
27	0E-M	-	-
28-29	30EF-D,30DE-M	11/13	+0
30	30E-M	-	-
31-32	0PF-D, 0DP-M	11/13	+1
33	0P-M	-	-
34-35	30PF-D,30DP-M	11/13	+1
36	30P-M	-	-
37-40	0DP-1, 0DP-3, 0DP-5, 0DP-7	11/13	+3
41	0DP-M	11/13	+3
42	0P-M	-	-
43-44	0EF-D, 0DE-M	11/13	+2
45	0E-M	-	-

Table 5.8. TED data—record timing, for record types 1, 2, & 3 (continued)

TED Element	Data channel	Accumulation period (s)	Accumulation begin time relative to T0 (s)
46-47	30EF-D,30DE-M	11/13	+2
48	30E-M	-	-
49-50	0PF-D, 0DP-M	11/13	+3
51	0P-M	-	-
52-53	30PF-D,30DP-M	11/13	+3
54	30P-M	-	-
55-58	30DP-1, 30DP-3, 30DP-5, 30DP-7	11/13	+5
59	30DP-M	11/13	+5
60	30P-M	-	-
61-62	0EF-D, 0DE-M	11/13	+4
63	0E-M	-	-
64-65	30EF-D,30DE-M	11/13	+4
66	30E-M	-	-
67-68	0PF-D, 0DP-M	11/13	+4
69	0P-M	-	-
70-71	30PF-D,30DP-M	11/13	+4
72	30P-M	-	-

TED Background values, 0E-BK, 30E-BK, 0P-BK, and 30E-BK, are listed in place of the energy flux values for sweep intervals 1, 3, 5, and 7 in RECORD TYPE 4. Table 5.9 shows where the background data appear in the record. (Note that background data are given only in RECORD TYPE 4.)

Table 5.9. TED background data channels – record type 4

TED Element	Detector
1- 4	OE-BK, 30E-BK, 0P-BK, 30P-BK
5-18	same as RECORD TYPES 1, 2, & 3
19-22	No Data
23-36	same as RECORD TYPES 1, 2, & 3
37-40	No Data
41-54	same as RECORD TYPES 1, 2, & 3
55-58	No Data
59-72	same as RECORD TYPES 1, 2, & 3

TED Background data are the total number of counts accumulated during the first 1/13 s (background interval) of each of the 16 energy sweeps for each of the detectors, summed and read out once every 32 s.

The TED flux values are shown in Table 5.10. These values are calculated and are not telemetered data.

Table 5.10. TED Total Energy Flux Values

TED Flux Element	8-s interval
1	first 2-s experiment cycle
2	second 2-s experiment cycle
3	third 2-s experiment cycle
4	fourth 2-s experiment cycle

6.0. Archive Tapes

The Space Environment Laboratory (SEL) receives raw-telemetry data from the TIROS/NOAA satellite tracking sites in near-real time. Every 10 days the raw-telemetry data are processed and reformatted. Every month the data are written onto archive tapes in a binary format. The archive processing adds information, calculates values, reformats the raw-telemetry data, and packs all information into 8-second logical records. Figure 6.1 is a printout of all the information found in one 8-second logical record.

A logical record contains

- Orbital information
- Total Energy Flux values calculated from TED instrument data
- Spacecraft housekeeping information
- Spacecraft ID and timing information
- Instrument status and calibration status flags
- 8 seconds of MEPED instrument data
- 8 seconds of HEPAD instrument data
- 8 seconds of TED instrument data

Logical records are 332-byte, binary records; they are grouped into files, each of which contains 8 to 11 days of data. Three data files are created each calendar month; usually, only about 2 percent of all possible data are missing.

- The first data file of the month contains data from days 1 through 10.
- The second data file of the month contains data from days 11 through 20.
- The third data file of the month contains data from day 21 through the end of the month (day 28 to 31, depending on the month).

SEL writes the TIROS/NOAA archive data on 3480 magnetic tapes. The tapes are unlabeled and in a non-ansi tape format, they are written using the unix command *data dump* (*dd*). Each 3480 tape contains five files: two ascii files, to help in restoring, unpacking, and using the data; and three binary data files. Appendix A contains sample unix commands to restore data files from a 3480 tape as well as sample programs to read data records.

TIROS/NOAA Archive Tapes written by SEL:

File 1. Header file, ascii file, 80-byte blocks

File 2. Unpack routines, ascii file, 80-byte blocks

File 3. First data file (days 1 through 10), binary file, 32768-byte blocks—about 35 mbytes.

File 4. Second data file (days 11 through 20), binary file, 32768-byte block—about 35 mbytes.

File 5. Third data file (days 21 through end of month), binary file, 32768-byte blocks—maximum 40 mbytes.

Note: tapes distributed by NGDC have an additional ascii file, at the front of the tape, containing NGDC's tape identification number. All other files are identical.

```
TIROS/NOAA ARCHIVE RECORD FOR YEAR  94    DAY   70        HOUR   1       MIN  50    MSEC 37500
S/C ID  5     YEAR 94    DAY 70     MS 6637500     REC STA  0     AOLT  8150   INCL  987   ORBIT  14557   REC TYPE  2
   SAT PAR  GEOG LAT   61.27    LONG  247.85    BR  -40126    BT    -5738    BP     2544    BB    40615
  FOFL PAR  GEOG LAT   62.08    LONG  248.70    BR  -56678    BT    -6420    BP     3641    BB    57157
            GEOM LAT   69.38    LONG  299.53    L VALUE  8.74    TED0  10.56    TED30  44.96
            MEP81   87.07    MEP83   89.06    MEP0    8.89    L TIME 275.50    ML TIME 262.86    VERS  2

      ON/OFF     MEPED 1    HEPAD 0    TED 1     IFC    MEPED 0    TED/HEPAD 0    TED MODE 0   FORMAT 1   TED PHD  0

              MPTT    METT    MELT    OMNI    AMSS   HELT    PMT    PMHV   HSSD  LVL   TEPS   TPSS   LVR   CEA    TEDT
HOUSEKEEPING -16.7   -16.7   -16.7   -30.6    0.0   -66.9   -66.9   0.7    1.0  0.0    0.0    0.0   3.0  456.0  -16.7

          HEPAD DATA IN COUNTS
     BLOCK        P1       P2       P3       P4       A1       A2       S5       S4       S1       S2       S3
         1         0        0        0        0        0        0        0        0        0        0        0
         2         0        0        0        0        0        0        0        0        0        0        0

          MEPED DATA      IONS    0I     -1    90I    -1
SUB  0P1   0P2   0P3   0P4   0P5   0E1   0E2   0E3  90P1  90P2  90P3  90P4  90P5  90E1  90E2  90E3  P6  P7 P8
5     42     0     0     0     0  2113    28     0    20     1     0     0     0  1697    24     1   4   3  2
6     60     2     0     0     0   213     3     0    15     2     0     0     0    26     1     0   2   2  1
7     29     1     1     0     0    24     1     0    12     0     0     0     0    19     2     1   2   4  0
8     18     3     0     0     0    12     0     0    18     0     0     0     0    20     2     0   2   4  4

   0DE1   0DE3   0DE5   0DE7    0DE    0EM     FLUX      30DE1  30DE3  30DE5  30DE7  30DEM  30EM      FLUX
    689   2753   2113  11009  11009     7    60.741       377    561   1889   5505   6273     6    26.454

   0DP1   0DP3   0DP5   0DP7    0DP    0PM     FLUX      30DP1  30DP3  30DP5  30DP7  30DP   30PM      FLUX
      0      2      1      0      2     3     0.749         0      0      1      0      0     2     0.076

      0EFD     0DEM      0EM    30EFD    30DEM     30EM    0PFD    0DPM     0PM    30PFD   30DPM    30PM
    9985.0    11009        7   9473.0    13057        7     3.6       2       1      1.4       2       2
    3905.0     6529        7   4225.0     6273        6     1.3       2       3      4.8       2      10
     213.0     1313        3     95.0      625        3     1.7       2       3      1.1       2       4
      11.5       16        3      9.5     ··19        2     1.9       0       3      2.9       0       2
```

Figure 6.1. A TIROS/NOAA Archive logical record.

File 1—Header file

The Header file is a copy of the commands used to write this archive tape. It shows how many files are on the tape and the dates of the data. SEL's convention for TIROS/NOAA data filenames is

```
Nsyddd.NEW     where:
N     designates NOAA, always N
s     last digit of the satellite number, for example 6 = NOAA-6,
        see page 5-2 for other numbers
y     last digit of the year of the data
ddd   day of the year of first data in the file
NEW   always NEW.
```

A sample Header file

```
set -x
echo "Processing Tape N62152-N62172. 1-30 Jun 1982. Sat 6"
/bin/mt -f /dev/rmt3h rewind
dd if=/tiros2/3480_write.1 of=/dev/nrmt3h obs=80
dd if=/tiros2/unpack2.c    of=/dev/nrmt3h obs=80
dd if=/tiros2/N62152.NEW   of=/dev/nrmt3h obs=32768
dd if=/tiros2/N62162.NEW   of=/dev/nrmt3h obs=32768
dd if=/tiros2/N62172.NEW   of=/dev/rmt3h obs=32768
/bin/mt -f /dev/rmt3h rewind
/bin/mt -f /dev/rmt3h offline
```

File 2—Open and Unpack Routines

SEL provides two routines that open a file and then read and unpack the data records. These routines are written in ansi C but can be called from either C or Fortran programs. The routines, called *open_archive* and *unpack2*, perform the following functions.

open_archive ()

- Opens the file for reading and makes the filename global to the read programs. The name of the file to open is requested from the user.

unpack2 (End of File flag returned)

- Reads one 8-second logical record from the data file.
- Recognizes machine-specific word formatting (*little endian* .vs. *big endian*) to correctly interpret and store the data.
- Unpacks the logical record and converts it to engineering units.
- Writes the logical record into COMMON /REC/ for Fortran access, or structure rec_ for C access.

Files 3 to 5—Data files

Each data file is a binary file with 32768-byte blocks; 10-day files are about 35 mbytes long, and 11-day files are a maximum of 40 mbytes. After a data file is restored to disk it can be read with the *open_archive* and *unpack2* routines. The read routine, *unpack2*, puts the next logical record into a Fortran COMMON BLOCK or C structure.

Sections 6.1 through 6.8 describe the data and other information contained in each logical record.

6.1. Logical Records

Logical records contain 8 seconds of instrument data plus the related orbital and housekeeping information. The records are ordered by time. A time, T0, is associated with every record and is in milliseconds of the day. Orbital and housekeeping information are valid at T0, but instrument data are timed relative to T0. (See the discussion on record timing in Section 5.0.)

The original raw-telemetry data are received in 32-s records, but these long records are divided into four 8-s logical records during the archive processing. As the 8-s records are extracted from the 32-s raw record they are numbered, from 1 to 4, and these numbers are stored in the 8-s record as the "record type." The record type is important because certain fields in the MEPED and TED instrument data are not the same in all record types (all other information is the same in all record types). Note that because the four 8-s logical records come from a single record in the raw data they are always in consecutive order; therefore, time gaps are only possible between record types 4 and 1.

Every 332-byte logical record can be thought of as containing 8 kinds of data plus a program version number:

Table 6.1. Logical record format

Section	Data	Number of bytes	Byte Count
1	Orbital information	92	1–92
2	Total energy flux values	16	93–108
3	Spacecraft housekeeping information	30	109–138
4	Spacecraft ID and timing information	16	139–154
5	Instrument status and calibration flags	4	155–158
6	MEPED data	78	159–236
7	HEPAD data	22	237–258
8	TED data	72	259–330
	Program version number	1	331

These eight kinds of data are described below in Sections 6.1.1 through 6.1.8. Each section contains a table with information needed to use the data as they are written in the COMMON BLOCK (and C structure); the table also contains the information needed to unpack the logical record without using SEL's *upack2* routine:

Array Name and index—The unpack routine puts a logical record into a Fortran COMMON BLOCK or C structure in arrays. The heading of the first column shows the name of the array, and the first column in the table shows the index in the Fortran array. Of course, the index in the C structure array is one less, and for 2-dimensional arrays the indices are reversed.

Description or Data Channel—A brief description of the value or the data channel name.

Range—The minimum and maximum values of a variable after conversion.

Units—The units of the variable after conversion.

Conversion Factor—The number by which a data value from the disk file must be multiplied to convert it to usable units. **NOTE:** Not necessary if *unpack2* is used; values are converted to engineering units before being stored in the COMMON BLOCK or C structure.

The conversion factor for most data values is shown as "CC1" or "CC2." If the SEL routine *unpack2* is not used, the data values in the logical records can be converted to counts from two look-up tables (called CC1 and CC2) or by using a conversion algorithm. The look-up tables and conversion algorithms are described in Appendix A.

Number of bytes—The number of bytes used by the values in the disk file. An "S" means a signed value. (Not needed if *unpack2* is used.)

Byte count—The byte position of the data within the 332-byte logical record. (Not needed if *unpack2* is used.)

The program version number is written into the COMMON BLOCK as HEAD(23).

Appendix A shows the Fortran COMMON BLOCK and the C structure.

6.1.1. Orbital Information

Table 6.2 shows the orbital information in every 8-s data record. Notice The supplied application routines store the first word in a logical record, *milliseconds of the day*, in the IHD array (with the other timing words) while the orbital imformation is stored in the HEAD array.

Table 6.2. Orbital Information

HEAD	Description	Range	Units	Conversion factor	Number of bytes	Byte count
IHD(4)	Milliseconds of the day	0 to 86400000			4	1–4
Data at the satellite						
1	Geographic latitude	–90.00 to 90.00	degrees	0.01	4S	5–8
2	Geographic east longitude	0.00 to 360.00	degrees	0.01	4S	9–12
3	BR	–41000 to 43000	nT		4S	13–16
4	BT	–27000 to 10000	nT		4S	17–20
5	BP	–10000 to 10000	nT		4S	21–24
6	BB	20000 to 45000	nT		4S	25–28
Data at the Foot Of the Field Line (FOFL)						
7	Geographic latitude	–90.00 to 90.00	degrees	0.01	4S	29–32
8	Geographic east longitude	0.00 to 360.00	degrees	0.01	4S	33–36
9	BR120	–58000 to 65000	nT		4S	37–40
10	BT120	–36000 to 15000	nT		4S	41–44
11	BP120	–15000 to 15000	nT		4S	45–48
12	BB120	20000 to 45000	nT		4S	49–52
13	Geomagnetic latitude	–90.00 to 90.00	degrees	0.01	4S	53–56
14	Geomagnetic east longitude	0.00 to 360.00	degrees	0.01	4S	57–60
15	L–value (set to 0.0 if greater than 14.99)	0.93 to 14.99		0.01	4S	61–64
Pitch angles						
16	TED 0 deg detector*	0.00 to 180.00	degrees	0.01	4S	65–68
17	TED 30 deg detector*	0.00 to 180.00	degrees	0.01	4S	69–72
18	MEPED 81 deg Proton detector**	0.00 to 180.00	degrees	0.01	4S	73–76
19	MEPED 83 deg Electron detector**	0.00 to 180.00	degrees	0.01	4S	77–80
20	MEPED 0 deg Electron detector**	0.00 to 180.00	degrees	0.01	4S	81–84
Miscellaneous information						
21	Local time**	0.00 to 360.00	degrees	0.01	4S	85–88
22	Magnetic local time**	0.00 to 360.00	degrees	0.01	4S	89–92
				Total bytes	92	

* at the Foot Of the Field Line (FOFL), 120 km
** at the satellite
Sensor angle descriptions are with respect to the zenith (local Earth vertical).

6.1.2. Total Energy Flux Values

Table 6.3 shows the Total Energy Flux Values in every 8-second record. These values were calculated from data from the Total Energy Detector (TED) instrument; see chapter 5.5 for more details.

Table 6.3. Total energy flux values

TEDFX	Description	Range	Units	Conversion factor	Number of bytes	Byte count
TED Flux						
1	Total Energy Flux	0 to 1000	mW/m^2	0.001	4	93–96
2	Total Energy Flux	0 to 1000	mW/m^2	0.001	4	97–100
3	Total Energy Flux	0 to 1000	mW/m^2	0.001	4	101–104
4	Total Energy Flux	0 to 1000	mW/m^2	0.001	4	105–108
				Total bytes	16	

6.1.3. Housekeeping Information

Table 6.4 shows the spacecraft housekeeping information for every 8-second record. See Table 5.2 for a description of the acronyms used below.

Table 6.4. Housekeeping format

HOUS	Description	Range	Units	Conversion factor	Number of bytes	Byte count
1	MPTT	−67.0 to 40.6	degrees C	0.1	2S	109–110
2	METT	−67.0 to 40.6	degrees C	0.1	2S	111–112
3	MELT	−67.0 to 40.6	degrees C	0.1	2S	113–114
4	OMNI	−81.0 to 4.9	degrees C	0.1	2S	115–116
5	AMSS	−0.00 to 4640.30	volts	0.01	2	117–118
6	HELT	−67.0 to 40.6	degrees C	0.1	2S	119–120
7	PMT	−67.0 to 40.6	degrees C	0.1	2S	121–122
8	PMHV	0.00 to 5.10	volts	0.01	2	123–124
9	HSSD	0.0 to 5478.4	volts	0.1	2	125–126
10	LVL	0 to 128*	level	1	2	127–128
11	TEPS	0 to 8*	level	1	2	129–130
12	TPPS	0 to 8*	level	1	2	131–132
13	LVR	0.00 to 5.10	volts	0.01	2	133–134
14	CEA	0.0 to 791.5	volts	0.1	2	135–136
15	TEDT	−67.0 to 40.6	degrees C	0.1	2S	137–138
				Total bytes	30	

* The valid range begins at 1 and bad data are denoted by a zero (0).

6.1.4. Year, Day, and Spacecraft Information

Table 6.5 defines the year, day and other spacecraft information in every 8-second record. Note: T0, the record time includes year and day which are shown below plus the milliseconds of the day which are shown in Table 6.2.

Table 6.5. Year, day, and spacecraft informaiton format

IHD	Description	Range	Units	Number of bytes	Byte count
1	Spacecraft ID	0 to 9		2	139–140
2	Year	0 to 99		2	141–142
3	Day of the year	1 to 366		2	143–144
5	Receiving station	0 to 6		2	145–146
6	Altitude	about 850.0	kilometers	2	147–148
7	Inclination	about 99.0	degrees	2	149–150
8	Orbit number (>12 years)	1 to 65535		2	151–152
9	Record type	1 to 4		2	153–154
			Total bytes	16	

6.1.5. Instrument Status and Calibration Flags

Table 6.6 gives the instrument status information, including calibration flags, in every 8-s data record.

Table 6.6. Instrument status and calibration flags format

ISTAT	Description	Status	Position	Bits	Number of bytes	Byte count
1	MEPED on/off	1 = on, 0 = off	MSB**	1		
2	HEPAD on/off	1 = on, 0 = off		1		
3	TED on/off	1 = on, 0 = off		1		
4	MEPED IFC*	1 = yes, 2 = no		1		
5	TED/HEPAD IFC*	1 = yes, 2 = no		1		
6	TED mode	0 to 3		1		
7	Telemetry format	1 = 1, 0 = 2	LSB**	1	2	154–155
8	TED PHD flags	0 to 15		16	2	157–158
		Total bytes			4	

* IFC = In-Flight Calibration

** MSB = the most significant bit of the word; LSB = the least significant bit.

6.1.6. MEPED Data

Table 6.7 shows the MEPED data in one 8-s data record. The 0° and 90° $Z \geq 2$ ions (MEPI words 1 and 2) are read out only once every 16 seconds and are therefore present only in record types 1 and 3. Words 1 and 2 of record types 2 and 4 contain minus 1 (−1). The record type is in the orbital information in every data record. All other channels are read every 2 seconds and are in four groups within the 8-s record. Remember that "counts" in the data record are counts per accumulation period.

Table 6.7. MEPED data format

MEPI	Data channel	Units	Conversion factor	Number of bytes	Byte count
	Ions				
1	0I (Record types 1 & 3 only)	counts	CC1	1	159
2	90I (Record types 1 & 3 only)	"	"	1	160

MEP	Data channel	Units	Conversion factor	Number of bytes	Byte count
	Protons and Electrons				
1 to 5	0P1 to 0P5	"	"	5 X 1	161–165
6 to 8	0E1 to 0E3	"	"	3 X 1	166–168
9 to 13	90P1 to 90P5	"	"	5 X 1	169–173
14 to 16	90E1 to 90E3	"	"	3 X 1	174–176
17 to 19	P6 to P8	"	"	3 X 1	177–179
repeat words 1 to 19 (0P1 to P8)		"	"	19 X 1	180–198
repeat words 1 to 19 (0P1 to P8)		"	"	19 X 1	199–217
repeat words 1 to 19 (0P1 to P8)		"	"	19 X 1	218–236
		Total bytes		**78**	

6.1.7. HEPAD Data

Table 6.8. shows the HEPAD data in one 8-s data record. Each data channel is read every 2 seconds, and the data are in four groups within the 8-s record. Remember that "counts" in the data record are counts per accumulation period.

Table 6.8. HEPAD data format

IHEP	Data channel	Units	Conversion factor	Number of bytes	Byte count
1	P1	counts	CC1	1	237
2	P2	"	"	1	238
3	P3	"	"	1	239
4	P4	"	"	1	240
5	alpha 1	"	"	1	241
6	alpha 2	"	"	1	242
7	S5	"	"	1	243
8	S4	"	"	1	244
9	S1	"	"	1	245
10	S2	"	"	1	246
11	S3	"	"	1	247
	repeat P1 to S3	"	"	11 X 1	248–258
			Total bytes	22	

6.1.8. TED Data

The TED data set is much larger and more complex than the other instrument data sets. It contains data from four different detectors, and each detector is sampled four times in every 8-s data record. However, although some values are read out four times per record, others are included only once. Table 6.9 shows the TED record format in record types 1, 2, and 3. Table 6.10 shows record type 4, which contains background information as well as the repeated detector data. Note theseTables are for operating mode 0 or 2; the only configurations in which the TED is normally operated.

Each data record contains TED data from four detectors: the 0° electron detector, the 30° electron detector, the 0° proton detector, and the 30° proton detector. In record types 1, 2, and 3, the TED data contain four groups of 18 data words. The first six words of each group are different energy-interval readings from one detector; each of the four detectors is used in one of the four groups. Words 7-18 are the same in each of the four groups but contain only the maximum energy interval reading from each detector. In record type 1 the first 18 words contain all zeros: at the beginning of the tape, after a time gap, or if the preceding record is missing.

The TEDFX array contains the four integrated energy flux values calculated from the TED sensor responses in each logical record. The largest possible valid energy-flux value is approximately 600 mW/($m^2 \cdot s$). A TEDFX value of 1000 denotes that bad or noisy data were identified within that 2-s data set (see Appendix C2). A TEDFX value of 995 indicates the TED instrument was not operating in its normal mode (see Section 5.2). This occurred from time to time during the lives of NOAA−6 and TIROS−N.

In record type 4, words 1−4 of the first group contain background data. The first four words of the following groups contain zeros. Words 5−18 of all groups are the same as in record types 1, 2, and 3 (see Table 6.10).

TED data are converted to counts using the CC1 or CC2 tables (except, of course, the Total Energy Flux values and 0E−M, 30E−M, 0P−M, and 30P−M, which need no conversion).

Table 6.9. TED data format—record types 1, 2, and 3

TED	Data channel	Range	Units	Conversion factor	Number of bytes	Byte count
1	0DE−1		counts	CC1	1	259
2	0DE−3		"	"	1	260
3	0DE−5		"	"	1	261
4	0DE−7		"	"	1	262
5	0DE−M		"	"	1	263
6	0E−M	1−11			1	264
7	0EF−D		"	CC2	1	265
8	0DE−M		"	CC1	1	266
9	0E−M	1−11			1	267
10	30EF−D		"	CC2	1	268
11	30DE−M		"	CC1	1	269
12	30E−M	1−11			1	270
13	0PF−D		"	CC2	1	271
14	0DP−M		"	CC1	1	272
15	0P−M	1−11			1	273
16	30PF−D		"	CC2	1	274
17	30DP−M		"	CC1	1	275
18	30P−M	1−11			1	276
			total bytes in group		18	
	30DE−1, 3, 5, 7		counts	CC1	4	277−280
	30DE−M		"	CC1	1	281
	30E−M	1−11			1	282
repeat 0EF−D to 30P−M (words 7−18)					12	283−294
			total bytes in group		18	
	0DP−1, 3, 5, 7		counts	CC1	4	295−298
	0DP−M		"	CC1	1	299
	0P−M	1−11			1	300
repeat 0EF−D to 30P−M (words 7−18)					12	301−312
			total bytes in group		18	
	30DP−1, 3, 5, 7		counts	CC1	4	313−316
	30DP−M		"	CC1	1	317
	30P−M	1−11			1	318
repeat 0EF−D to 30−PM (words 7−18)					12	319−330
			total bytes in group		18	
			Total bytes		72	

Table 6.10. TED background data format—record type 4

TED	Data channel	Range	Units	Conversion factor	Number of bytes	Byte count
1	0E–BK		counts	CC1	1	259
2	30E–BK		"	"	1	260
3	0P–BK		"	"	1	261
4	30P–BK		"	"	1	262
5–18	same as record types 1, 2 & 3					263–276
			total bytes in group		**18**	
	same as record types 1, 2, & 3					227–294
			total bytes in group		**18**	
	same as record types 1, 2, & 3					295–312
			total bytes in group		**18**	
	same as record types 1, 2, & 3					313–330
			total bytes in group		**18**	
			Total bytes		**72**	

Note: The format of TED Total Energy Flux data is shown in table 6.3.

Table 6.10. FED background data format—record type 4

Field	Name	Range	Units	Conversion factor	Number of bytes	Byte count
[illegible]	[illegible]		[illegible]	[illegible]	1	259
[illegible]	[illegible]				1	260
[illegible]	[illegible]				1	261
[illegible]	[illegible]				1	262
[illegible]	group [illegible] record types 1, 2, & 3					263–276
		total bytes in group			14	
	group [illegible] record types 1, 2, & 3					277–294
		total bytes in group			18	
	[illegible] record types 1, 2, & 3					295–312
		total bytes in group			18	
	[illegible] record types 1, 2, & 3					313–330
		total bytes in group			18	
		Total bytes			72	

[a] See [illegible] FED [illegible] data shown in table 6.3

APPENDIX A: Reading TIROS/NOAA Archive Tapes

A.1. Restoring data from 3480 tape to disk

The following illustrative UNIX command sequence shows how the five files from a 3480 archive tape are restored to disk files (# is the drive number).

mt −f /dev/nrmt#h rewind	mounts the tape
dd if=/dev/nrmt#h of=information_file ibs=80	copies first file into information_file
dd if=/dev/nrmt#h of=unpack2.c ibs=80	copies second file into unpack2.c
dd if=/dev/nrmt#h of=N23001.NEW ibs=32768	copies third file into N23001.NEW
dd if=/dev/nrmt#h of=N23011.NEW ibs=32768	copies fourth file into N23011.NEW
dd if=/dev/nrmt#h of=N23021.NEW ibs=32768	copies fifth file into N23021.NEW
/mt −f /dev/rmt#h rewind	rewinds the tape

A.2. Using the SEL provided routines

Table A.1 shows the elements of the common block /REC/ and the C structure.

NOTE: In record type 4, TED(1,1) through TED(4,1) contain background counts, and TED(1,n) through TED(4,n) for n = 2, 3, 4 contain no data.

If the previous record is missing, which can only be true when the record type is 1, TED (1,1) through TED (18,1) contain zeros. The previous record is missing for the first record on the tape and for the first record after a time gap.

SEL includes a file on each archive tape with two routines: *open_archive* and *unpack2*. These routines are designed to be used from either C or Fortran programs by defining either a C structure or a FORTRAN common block to hold the values of the archive record.

open_archive: Requests the user to input the name of the archive file on disk, and then opens that file.

unpack2: Reads the next logical record on the disk file, unpacks the data, converts telemetry data to engineering units, and finally writes the Fortran common block /rec/ or the C structure rec_. *Unpack2* takes one integer argument, *iend,* and sets *iend* = 0 when a record is read and written to /REC/ or sets *iend* = 1 when the end of the file is encountered. /REC/ remains unchanged from the last reading when the EOF is read.

From Fortran

```
CALL open_archive
CALL unpack2(IEND)
```

The following variables and the common block /REC/ must be defined. Each call to *unpack2* returns the values for one logical record in the common block /REC/. Both REAL and INTEGER values are 4 bytes.

```
INTEGER*4        IEND
REAL*4           HEAD, HOUS, TED, TEDFX
INTEGER*4        IHD, ISTAT, MEPI, MEP, IHEP
COMMON /REC/     IHD(9), HEAD(23), ISTAT(8), HOUS(15), MEPI(2),
                 MEP(19,4), IHEP(11,2), TED(18,4), TEDFX(4)
```

From C

```
open_archive_();
unpack2_(&iend);
```

In C, a structure must be declared to hold the data for one logical record. This structure definition is case sensitive. Note that the name rec_ is lower case and is followed by the underscore character. This is the convention required to pass data between C and FORTRAN. Even though the unpacking routine is written in C, this convention is necessary because the unpacking routine is written to enable FORTRAN routines to access the data. Both long int and float data types are 4 bytes.

```
extern struct archive_rec {
        long int IHD[9];
        float HEAD[23];
        long int ISTAT[8];
        float HOUS[15];
        long int MEPI[2];
        long int MEP[4][19];
        long int IHEP[2][11];
        float TED[4][18];
        float TEDFX[4];
} rec_;
long int iend;
```

Table A.1. Contents of COMMON /REC/ IHD(9), HEAD(23), ISTAT(8), MEPI(2), MEP(19,4), IHEP(11,2), TED(18,4), TEDFX(4)

Variable	Type	Description
ORBITAL INFORMATION		
IHD(1)	Integer	Spacecraft ID
IHD(2)	"	Year
IHD(3)	"	Day of the year
IHD(4)	"	Millisecond of the day
IHD(5)	"	Receiving station
IHD(6)	"	Altitude
IHD(7)	"	Inclination
IHD(8)	"	Orbit number
IHD(9)	"	Record type
HEAD(1)	Floating point	Geographic latitude at the satellite
HEAD(2)	"	Geographic east longitude at the satellite
HEAD(3)	"	BR at the satellite
HEAD(4)	"	BT at the satellite
HEAD(5)	"	BP at the satellite
HEAD(6)	"	BB at the satellite
HEAD(7)	"	Geographic latitude at the FOFL
HEAD(8)	"	Geographic east longitude at the FOFL
HEAD(9)	"	BR120 at the FOFL
HEAD(10)	"	BT120 at the FOFL
HEAD(11)	"	BP120 at the FOFL
HEAD(12)	"	BB120 at the FOFL
HEAD(13)	"	Geomagnetic latitude at the FOFL
HEAD(14)	"	Geomagnetic east longitude at the FOFL
HEAD(15)	"	L−value
HEAD(16)	"	TED0 Pitch Angle
HEAD(17)	"	TED30 Pitch Angle
HEAD(18)	"	MEPED81 Pitch Angle
HEAD(19)	"	MEPED83 Pitch Angle
HEAD(20)	"	MEPED0 Pitch Angle
HEAD(21)	"	Local time at the satellite
HEAD(22)	"	Magnetic Local time at the satellite
HEAD(23)	"	Program version
HOUSEKEEPING INFORMATION		
ISTAT(1)	Integer	MEPED on/off
ISTAT(2)	"	HEPAD on/off
ISTAT(3)	"	TED on/off
ISTAT(4)	"	MEPED IFC
ISTAT(5)	"	TED/HEPAD IFC
ISTAT(6)	"	TED MODE
ISTAT(7)	"	TELEMETRY FORMAT
ISTAT(8)	"	TED PHD flags
HOUS(1) − HOUS(15)	Floating point	Instrument housekeeping words

Table A.1. continued

Variable	Type	Description
MEPED DATA		
MEPI(1), MEPI(2)	Integer	0I, 90I
MEP(1,n) – MEP(5,n)	"	0P1 to 0P5
MEP(6,n) – MEP(8,n)	"	0E1 to 0E3
MEP(9,n) – MEP(13,n)	"	90P1 to 90P5
MEP(14,n) – MEP(16,n)	"	90E1 to 90E3
MEP(17,n) – MEP(19,n)	"	P6 to P8
where n = 1 to 4		
HEPAD DATA		
IHEP(1,n) – IHEP(4,n)	Integer	P1 to P4
IHEP(5,n)	"	alpha 1
IHEP(6,n)	"	alpha 2
IHEP(7,n)	"	S5
IHEP(8,n)	"	S4
IHEP(9,n)	"	S1
IHEP(10,n)	"	S2
IHEP(11,n)	"	S3
where n = 1 to 2		
TED DATA		
TED(1,j)	Integer	data channel 1
TED(2,j)	"	data channel 3
TED(3,j)	"	data channel 5
TED(4,j)	"	data channel 7
TED(5,j)	"	maximum interval
TED(6,j)	"	count in maximum interval
where j=1,4		
1 = values for 0 degree electron detector		
2 = values for 30 degree electron detector		
3 = values for 0 degree proton detector		
4 = values for 30 degree proton detector		
TED(7,n) – TED(9,n)	"	0EF-D, 0DE-M, 0E-M
TED(10,n) – TED(12,n)	"	30EF-D, 30DE-M, 30E-M
TED(13,n) – TED(15,n)	"	0PF-D, 0DP-M, 0P-M
TED(16,n) – TED(18,n)	"	30PF-D, 30DP-M, 30P-M
TEDFX(n)	Floating point	Total Energy Flux
where n = 1 to 4, for each 2 s sample in the 8 s record.		

A.3. Sample C Program

Program *prarch* is an example of printing an 8-s logical records from a disk file in the one page per record format shown in figure 6.1. Only the main routine, print routine, and two time conversion routines are shown.

prarch.c

```
/***********************************************************************
 * FILE:        prarch.c
 *
 * DESCRIPTION:Main driver to print TIROS/NOAA archive records.
***********************************************************************/
#include <stdio.h>
#include <string.h>
/*** structure for one logical archive data record***/
extern struct archive_rec {
        long int  IHD[9];
        float     HEAD[23];
        long int  ISTAT[8];
        float     HOUS[15];
        long int  MEPI[2];
        long int  MEP[4][19];
        long int  IHEP[2][11];
        float     TED[4][18];
        float     TEDFX[4];
} rec_;

/***********************************************************************
 * ROUTINE:    main (int argc, char *argv[])
 *
 * DESCRIPTION:Print TIROS/NOAA archive records:
 *             Open the input disk file specified in arg[1].
 *             Open the output file specified in arg[2].
 *             Skip first n records, where n is specified in arg[3].
 *             Read and write formatted logical records to the output file
 *                until record m, where m is specified in arg[4].
 *        This code runs on a DEC 5000 workstation.
 *
 * PARAMETERS:  prtarch expects four input parameters:
 *              Input file name
 *              Output file name
 *              Start record number    (Counting from 1 as first record)
 *              End record number
 *              Example:  prtarch input_name output_name 1 200
***********************************************************************/
main(int argc, char *argv[])
{
FILE *input_fp, *output_fp;
long count;
long startRecord, endRecord;
long int IEOF;
/*** if user wants help, or does not input the required number of parameters -
        print messages and quit ***/
```

```
if (argc < 5 || strcasecmp(argv[1],"HELP") == 0) {
     printf("The parameters to prtarch are:\n");
     printf("1.     Input file name\n");
     printf("2.     Output file name\n");
     printf("3.     Start record number\n");
     printf("4.     End record number\n");
     printf("ex.  prtarch input_name output_name 1 200\n");
     exit(0);
}
/*** get start and end record numbers from input parameters ***/
startRecord = atoi(argv[3]);
endRecord = atoi(argv[4]);
count = 0;
IEOF = 0;

/*** open archive data file ***/
        /*** This is a modified version of open_archive that passes the
             filename as an argument instead of requesting user input ***/
open_archive_(argv[1]);

/*** open output file ***/
        /*** A open a common ascii output file  ***/
open_output_file_(argv[2]);

/*** read to first record to print using the standard SEL unpack2 routine ***/
while (!IEOF && count < startRecord) {
     unpack2_(&IEOF);
     count++;
}

/*** read and write requested records ***/
while (!IEOF && count < endRecord) {
     count++;
     unpack2_(&IEOF);
     print_archive_record_();
     }

}    /* End of main */
```

print_archive_record.c

```
/*************************************************************************
 * FILE:      print_archive_record.c, contains: print_archive_record_ *
 * DESCRIPTION:     writes the logical archive record currently in the
 *                  external struct rec_ to the output file.
*************************************************************************/

#include <stdio.h>
#include "proto.h"

static FILE *fp;    /*** declare global output file pointer ***/
/*************************************************************************
 * FUNCTION:        void print_archive_record_()
 *
 * DESCRIPTION:     writes all the values in a logical archive record,
```

```
 *                  in struct rec_, to an output file as a formatted full page
 *************************************************************************/
void print_archive_record_()
{
/*** declare record to be printed as external to this file ***/
extern struct archive_rec {
        long int IHD[9];
        float HEAD[23];
        long int ISTAT[8];
        float HOUS[15];
        long int MEPI[2];
        long int MEP[4][19];
        long int IHEP[2][11];
        float TED[4][18];
        float TEDFX[4];
} rec_;
double          time, zero;
unsigned long year, doy, hour, min, ms;
int       i, j, k;
int       mepi0,mepi1;

/* convert time in ihd to year,doy, hour, min, and sec  */
zero = 0.;
time = input_to_tiros_time(rec_.IHD[1], rec_.IHD[2], zero,
          zero, rec_.IHD[3]);
tiros_to_input_time(time, &year, &doy, &hour, &min, &ms);
/* Date, time and satellite position */
fprintf(fp,"\n\nTIROS/NOAA ARCHIVE RECORD FOR YEAR %3ld\tDAY %4ld\t",
          year,doy);
fprintf(fp,"HOUR %3ld\tMIN %3ld\tMSEC%6ld\n",
          hour, min, ms);
fprintf(fp,"S/C ID %2ld\tYEAR %3ld\tDAY %4ld\tMS %10ld\tREC STA %4ld\t",
          rec_.IHD[0], rec_.IHD[1], rec_.IHD[2], rec_.IHD[3], rec_.IHD[4]);
fprintf(fp,"ALT %5ld\tINCL %4ld\t",
          rec_.IHD[5], rec_.IHD[6]);
fprintf(fp,"ORBIT %6ld\tREC TYPE %2ld\n",rec_.IHD[7], rec_.IHD[8]);

/* orbital information */
fprintf(fp,"   SAT PAR  GEOG LAT%7.2f   LONG%8.2f   BR%8.0f   BT%8.0f",
          rec_.HEAD[0], rec_.HEAD[1], rec_.HEAD[2], rec_.HEAD[3],
fprintf(fp,"   BP%8.0f   BB%8.0f\n",
          rec_.HEAD[4], rec_.HEAD[5]);
fprintf(fp,"  FOFL PAR  GEOG LAT%7.2f   LONG%8.2f   BR%8.0f   BT%8.0f",
          rec_.HEAD[6], rec_.HEAD[7], rec_.HEAD[8], rec_.HEAD[9]);
fprintf(fp,"   BP%8.0f   BB%8.0f\n",
          rec_.HEAD[10], rec_.HEAD[11]);
fprintf(fp,"                GEOM LAT%7.2f   LONG%8.2f   L VALUE%6.2f   ",
          rec_.HEAD[12], rec_.HEAD[13], rec_.HEAD[14]);
fprintf(fp,"   TED0%7.2f   TED30%7.2f\n",
          rec_.HEAD[15], rec_.HEAD[16]);
fprintf(fp,"                MEP81%7.2f   MEP83%7.2f   MEP0%7.2f   L TIME%7.2f",
          rec_.HEAD[17], rec_.HEAD[18], rec_.HEAD[19], rec_.HEAD[20]);
fprintf(fp,"   ML TIME%7.2f   VERS%3.0f\n\n",
```

```
        rec_.HEAD[21],rec_.HEAD[22]);
/*  status, and housekeeping words  */
fprintf(fp,"          ON/OFF    MEPED%2d   HEPAD%2d",
        rec_.ISTAT[0], rec_.ISTAT[1]);
fprintf(fp,"  TED%2d   IFC  MEPED%2d   TED/HEPAD%2d   ",
        rec_.ISTAT[2], rec_.ISTAT[3], rec_.ISTAT[4]);
fprintf(fp,"TED MODE%2d  FORMAT%2d   TED PHD%3d\n\n",rec_.ISTAT[5],
        rec_.ISTAT[6], rec_.ISTAT[7]);
fprintf(fp,"                         MPTT    METT    MELT    OMNI     ");
fprintf(fp,"AMSS    HELT     PMT     PMHV    HSSD     LVL     TEPS    ");
fprintf(fp,"TPSS     LVR     CEA     TEDT\n");
fprintf(fp,"  HOUSEKEEPING");
for (i=0;i<15;i++)
    fprintf(fp,"%8.1f",rec_.HOUS[i]);
fprintf(fp,"\n");

/* hepad data */
fprintf(fp,"          HEPAD DATA IN COUNTS\n");
fprintf(fp,"    BLOCK       P1      P2      P3      P4      A1     ");
fprintf(fp,"A2      S5      S4      S1      S2      S3\n");
for (i=0;i<2;i++) {
    fprintf(fp,"       %2d",i+1);
    for (j=0;j<11;j++) {
        fprintf(fp,"%8d",rec_.IHEP[i][j]);
    }
    fprintf(fp,"\n");
}
/* meped data */
mepi0 = rec_.IHD[8] == 2 || rec_.IHD[8] == 4 ? -1 : rec_.MEPI[0];
mepi1 = rec_.IHD[8] == 2 || rec_.IHD[8] == 4 ? -1 : rec_.MEPI[1];
fprintf(fp,"          MEPED DATA     IONS   0I%5d   90I%5d\n",mepi0,mepi1);
fprintf(fp," SUB    0P1    0P2    0P3    0P4    ");
fprintf(fp," 0P5    0E1    0E2    0E3   90P1   90P2   ");
fprintf(fp,"90P3   90P4   90P5   90E1   90E2   90E3   P6   P7   P8\n");
j = (rec_.IHD[8] - 1) * 4;
for (i=0;i<4;i++) {
    fprintf(fp," %2d",j+i+1);
    for (k=0;k<16;k++)
        fprintf(fp,"%7d",rec_.MEP[i][k]);
    for (k=16;k<19;k++)
        fprintf(fp,"%5d",rec_.MEP[i][k]);
    fprintf(fp,"\n");
}
fprintf(fp,"\n");
/* ted data */
fprintf(fp,
    "       0DE1    0DE3    0DE5    0DE7    0DE     0EM        FLUX");
fprintf(fp,
    "      30DE1   30DE3   30DE5   30DE7   30DEM    30EM       FLUX\n");
for (i=0;i<2;i++) {
    fprintf(fp,"    ");
    for (j=0;j<6;j++)
        fprintf(fp,"%8.0f",rec_.TED[i][j]);
```

```
    fprintf(fp,"      %9.3f",rec_.TEDFX[i]);
}
fprintf(fp,"\n\n");
fprintf(fp,
    "          0DP1     0DP3     0DP5     0DP7      0DP      0PM          FLUX");
fprintf(fp,
    "         30DP1    30DP3    30DP5    30DP7     30DP     30PM          FLUX\n");
for (i=2;i<4;i++) {
    fprintf(fp,"      ");
    for (j=0;j<6;j++)
        fprintf(fp,"%8.0f",rec_.TED[i][j]);
    fprintf(fp,"      %9.3f",rec_.TEDFX[i]);
}
fprintf(fp,"\n");
fprintf(fp,"      0EFD     0DEM     0EM    30EFD    30DEM     30EM   ");
fprintf(fp,"     0PFD     0DPM     0PM    30PFD    30DPM    30PM\n");
for (i=0;i<4;i++) {
    fprintf(fp,
        "  %9.1f%8.0f%8.0f%9.1f%8.0f%8.0f%9.1f%8.0f%8.0f%9.1f%8.0f%8.0f\n",
        rec_.TED[i][6],rec_.TED[i][7],rec_.TED[i][8],rec_.TED[i][9],
        rec_.TED[i][10],rec_.TED[i][11],rec_.TED[i][12],
        rec_.TED[i][13], rec_.TED[i][14],rec_.TED[i][15],
    rec_.TED[i][16],rec_.TED[i][17]);
}
}    /* End of print_archive_record */
```

tiros_time.c

```
/********************************************************************
 * FILE:  tiros_time.c, contains input_to_tiros and tiros_to_input_time
 *
 * DESCRIPTION:    The date/time of data on the TIROS/NOAA Archive files is
stored as year, day of the year, and milliseconds of the day.
    Routine input_to_tiros_time converts these date/time variables to one
          variable - milliseconds since Jan 1, 1976 called TIME. TIME is an
          easy way to handle time within programs.
    Routine tiros_to_input_time converts TIME to Year, Day of the Year, Hour,
          Minute, and msec. The format used on the printout.
 ********************************************************************/
/********************************************************************
 *  ROUTINE:  input_to_tiros_time
 *
 *  DESCRIPTION:   converts Year, Day of the Year, Hour, Minute, Milliseconds
                   to Milliseconds since Jan 1, 1965.
               inputs - as long integers year, day of the year, hour, min,
                    msec (For Tiros date/time variables hour and min are set
                    to zero.)
               output - tiros time - double.
*********************************************************************/
double input_to_tiros_time(unsigned long year, unsigned long day,
        unsigned long hour, unsigned long min, unsigned long msec)
{
double        time;
unsigned long j;
```

```
time = 0.0;
if (year >= 78 && year <= 99) {
     j = year - 76;
     j = (j+3)/4 + 365*j + day - 1;
     time = 1440*j + (hour*60) + min + msec/60000.;
}
return(time);
}    /* End of input_to_tiros_time */

/********************************************************************
 *  ROUTINE:   tiros_to_input_time - converts milliseconds since Jan 1, 1976
 *               to Year, Day of Year, Hour, Minute, and Milliseconds.
 *
 *  PARAMETERS:     input - milliseconds since Jan 1, 1976 as a double
 *                  output -  year, hour, min, msec and long unsigned integers
 ********************************************************************/
void tiros_to_input_time(double time, unsigned long *year,
        unsigned long *day, unsigned long *hour, unsigned long *min,
        unsigned long *msec)
{
unsigned long  jday, j, j2;
long           day2;
day2 = 0;
jday = time / 1440;
*hour = (time - jday*1440) / 60;
*min = time - (jday*1440 + *hour*60);
*msec = 60000. * (time - (int)time);
j = jday / 365;
while (day2 <= 0) {
     j2 = (j+3) / 4;
     day2 = 1 + jday - (j*365 + j2);
     *year = 76 + j;
     j--;
}
*day = day2;
}    /* End of tiros_to_input_time */
```

A.4. Sample Fortran Program

```
        Program PRTUNP2
C-----------------------------------------------------
C PROGRAM: 'PRTUNP2.FOR' prints selected archive
C          data records per user request. 'PRTUNP2.FOR'
C          calls the c function 'unpack2.c'.
C
C PROGRAM USAGE:
C          Uses DEC Ultrix C-calling from FORTRAN convention.
C
C LIMITATIONS: N/A
C INPUT:   File dumped from 3480 tape. Filename 'Nxxxxx.NEW'
C OUTPUT:  'archive2.dat' contains archive data records printable
C          to an ASCII printer.
C COMPILATION:  f77 -c prtunp2.for
C LINKAGE:
C               (for DEC5000 ULTRIX system).
C               f77 -o prtunp2.exe prtunp2.o unpack2.o
C               where prtunp2.exe ------> output executable file,
C                     prtunp2.o   ------> object file from
C                                         'prtunp2.for' source,
C                     unpack2.o   ------> object file from
C                                         'unpack2.c' source
C
C PROGRAMMER: Minh Huynh
C REVISION: Revision 1.0. Sep 16, 1992
C
C-----------------------------------------------------
C       INTERFACE TO SUBROUTINE UNPACK_ [C] ()
        COMMON /rec/   IHD(9), HEAD(23), ISTAT(8),
     *              HOUS(15), MEPI(2), MEP(19,4),
     *              IHEP(11,2), TED(18,4), TEDFX(4)

        INTEGER*4 IREC, JREC, OLD_REC_NUM, NEW_REC_NUM
        CHARACTER*1 OLD_ANS, NEW_ANS
        INTEGER*2 YES, NO, IEND
        DATA YES/1/, NO/0/, JREC/0/, KREC/0/
        IEND = NO
C------------------------------------------------------------
C Open Archive file(s)
        CALL open_archive
        WRITE(*,4)
4       FORMAT('What starting record you want to print? ',$)
        READ(*,5) OLD_REC_NUM
5       FORMAT(I10)
        WRITE(*,6)
6       FORMAT('How many records you want to print? ',$)
        READ(*,5) ISIZE
        WRITE(*,7)
7       FORMAT ('Output file is named archive2.dat.')
        OPEN (UNIT = 7,FILE="archive2.dat",
     *        ACCESS='SEQUENTIAL', FORM='FORMATTED',
     *        STATUS= 'UNKNOWN')
```

```
C-----------------------------------------------------------------
C Write out the requested TIROS archive records
        KREC = 0
          DO WHILE (IEND .EQ. NO)
            CALL unpack2 (IEND)
             KREC = KREC + 1
              IF (KREC .GE. OLD_REC_NUM) THEN
                WRITE (7,10) KREC,
     *          (IHD(I),I=1,9),(HEAD(I),I=1,23),
     *          (ISTAT(I),I=1,8), (HOUS(I),I=1,15),(MEPI(I),I=1,2),
     *          ((MEP(I,J),I=1,19),J=1,4),
     *          ((IHEP(I,J),I=1,11),J=1,2),
     *          ((TED(I,J), I=1,18),J=1,4),
     *          (TEDFX(I),I=1,4)
              ENDIF
            IF (KREC .GT. OLD_REC_NUM + ISIZE) GOTO 100
          ENDDO
10        FORMAT(
     *       ' +--TIROS Data --------- Record:',i10,/,
     *       ' IHD(I)     = ',/,
     *       ' '     9(I10), /,
     *       ' HEAD(I)    = ',/,
     *                           6(F11.2,x),/,
     *                           6(F11.2,x),/,
     *                           6(F11.2,x),/,
     *                           5(F11.2,x),/,
     *       ' ISTAT(I) = ',/,
     *       '            ',     8(I5,2x), /,
     *       ' HOUS(I)    = ',/,
     *                           8(F7.2,x),/,
     *                           7(F7.2,x)/,
     *       ' MEPI(I)    = ',   2(I7,2x), /,
     *       ' MEP(I,J) = ',/,
     *                           7(10(I7,x),/),
     *                           6(I5,2x),/,
     *       ' IHEP(I,J) =',/,
     *                           2(11(I7,x),/),
     *       ' TED (I,J) =',/,
     *                          12(6(F12.4,x),/),
     *       ' TEDFX(I)   =',    4(F10.4,2x),/)
100     CONTINUE
C------------------------ Modified on May 1, 1993 mch ---------
        STOP
        END
```

Sample Run:

```
$ prtunp2.exe
 == TIROS / NOAA UNPACK2 Routine ==
 Enter the filename (e.g. N72141.NEW) :N01060.NEW
What starting record you want to print? 1
How many records you want to print? 1
Output file is named archive2.dat.
```

Sample Output:

```
$ more archive2.dat
 +--TIROS Data -------- Record:          1
 IHD(I)   =
          8      91     60   16732     0    8330     987   23103          1
 HEAD(I)  =
      33.13    282.95  -32013.00  -14921.00  -1136.00  35339.00
      35.84    282.65  -47963.00  -19504.00  -2006.00  51817.00
      47.07    351.74       2.32      30.85     51.73     84.50
      86.31     25.05     283.01     277.80      2.00
 ISTAT(I) =
             1      0      1      0      0      0      1      0
 HOUS(I)  =
 -20.40  -19.80  -17.20  -37.80   94.00  -66.90  -66.90   0.00
   0.00    0.00    1.00    1.00    3.00  660.30  -10.00
 MEPI(I)  =        0         0
 MEP(I,J) =
      1      1      0      0      0 303105 167937  17921      0      0
      0      0      0    593    103      7      3      0      3      0
      0      0      0      0 303105 167937  17921      2      0      1
      0      0    529     87      8      5      1      2      0      0
      0      0      0 303105 167937  16897      0      0      0      0
      0    593    107     12      1      3      2      0      0      0
      0      0 303105 167937  16129      1      0      1      0      0
   529    103      9      3      1      2
 IHEP(I,J) =
      1      0      0      0      0      0      0      0      0      0      0
      0      0      0      0      0      0      0      0      0      0      0
 TED (I,J) =
        9.0000        11.0000         6.0000         9.0000      1057.0000       11.0000
     1057.0000      1057.0000         0.0000      1057.0000      1057.0000        0.0000
        0.0625         0.0000        11.0000         0.0625         0.0000        1.0000
        5.0000         0.0000         0.0000         2.0000         5.0000        1.0000
       38.0000        11.0000         3.0000         7.7500         5.0000        1.0000
        0.0625         0.0000         3.0000         0.0625         0.0000        1.0000
        0.0000         0.0000         0.0000         0.0000         0.0000       10.0000
       42.0000        12.0000        10.0000         9.5000         4.0000        1.0000
        0.0625         0.0000        10.0000         0.0625         0.0000        1.0000
        0.0000         0.0000         0.0000         0.0000         0.0000        7.0000
       34.0000        12.0000         2.0000         4.2500         3.0000        7.0000
        1.9375         0.0000         2.0000         0.0625         0.0000        7.0000
 TEDFX(I)  = 1000.0000    1000.0000    1000.0000    1000.0000
```

A.5. upack2 and open_archive routines

The *unpack2* and *open_archive* routines as they are written on the TIROS/NOAA Archive Tapes.

```
/* =================================================================
   FUNCTION: unpack2.c

           'unpack2.c' function reads a buffer of TIROS/NOAA data
           which has been written in the SEL-specific format,
           unpacks the data and performs the table look-up to
           return engineering values to the calling function in c
           (or a FORTRAN subroutine.) The TIROS/NOAA data on the IBM-3480
           compatible tapes must be restored to disk files using the
           unix 'dd' command before it can be unpacked using 'unpack2.c'.

           The function has been written to be callable from either
           a C function or FORTRAN subroutine. The 'sys/types.h' and
           'netinet/in.h' header files are used to convert from BIG
           ENDIAN to LITTLE ENDIAN data format.

           'unpack2.c' was designed to allow for the transparent
           interface to existing SEL FORTRAN programs which incorporate
           the standard FORTRAN COMMON block '/REC'.

           The following machines and their internal data format
           representations have been tested using 'unpack2.c':

               Little ENDIAN (DEC, PC):          Bytes 1  2
               Big ENDIAN (Apollo, HP-UX):       Bytes 2  1

               Little ENDIAN (DEC, PC):          Bytes 1  2  3  4
               Big ENDIAN (Apollo, HP-UX):       Bytes 4  3  2  1


   CALLING FORMAT FROM FORTRAN:

           CALL unpack2 (IEND) where:
           (case-insensitive)
                                     IEND = 0 if End-of-File is not detected
                                          = 1 if End-of-File is detected.

   CALLING FORMAT FROM DEC 5000 Ultrix C:

           unpack2_ (long int &IEND);
           (case sensitive)

   COMPILATION:  c89 -c unpack2.c
                 Note: On the DEC 5000 Ultrix machine do not use the
                 f77 command to compile and link unpack2.c because
                 the DEC's preprocessor f77 will use cc instead c89.

   LINKAGE:    An example using FORTRAN program 'prtunp2.for' to link with
               the object of 'unpack2.c' follows:

               f77 -o prtunp2.exe prtunp2.o unpack2.o

               where:  'prtunp2.exe' -----> the output filename,
                       'prtunp2.o'   -----> object output of the FORTRAN
                                            example program
                                            which calls 'unpack2.c',
```

```
                'unpack2.o'   -----> object output of the
                                     'unpack2.c' function using
                                     the c89 compiler directive.

    PROGRAMMER: Minh Huynh

    REVISION: 1.0 -- Jan 22, 1993
    NOTES: Do NOT confuse this to 'unpack.c' routine which is used
           to unpack the TIROS/NOAA data buffers from the Cyber tapes.
    ================================================================== */
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <netinet/in.h>
#define DEC 1            /* change to 0 if used on machine with
                            default 16-bit int. e.g. IBM/PC-AT */
#if DEC==1
  typedef int L_int;
  typedef unsigned int u_L_int;
#endif
#if DEC==0
  typedef long int L_int;
  typedef unsigned long int u_L_int;
#endif

#define FOREVER (;;)

struct IN_P {
        long int  IHD_Msec;
        long int JHEAD[22];
    long int JTEDFX[4];
        short int JHOUS[15];
        short int IHD[8];
    short int ISTAT_P[2];
    unsigned char Index_MEPI[2];
    unsigned char Index_MEP[4][19];
    unsigned char Index_IHEP[2][11];
    unsigned char Index_TED[4][18];
    unsigned char Prog_Ver;
} in_p_;
/* sizeof_IN_P is 332 */
int sizeof_IN_P = 332;
int i=0, j=0, k=0, num_char;
struct S {
    long int IHD[9];
    float HEAD[23];
    long int ISTAT[8];
    float HOUS[15];
    long int MEPI[2];
    long int MEP[4][19];
    long int IHEP[2][11];
    float TED[4][18];
    float TEDFX[4];
    long int ISTAT_P[2];
```

```
    long int Index_MEPI[2];
    long int Index_MEP[4][19];
    long int Index_IHEP[2][11];
    long int Index_TED[4][18];
    long int IOPEN;
    long int IEND;
} r_;
struct archive_rec {
    long int IHD[9];
    float HEAD[23];
    long int ISTAT[8];
    float HOUS[15];
    long int MEPI[2];
    long int MEP[4][19];
    long int IHEP[2][11];
    float TED[4][18];
    float TEDFX[4];
} rec_;
char filename[] = "N23001.NEW";

FILE *ff_in, *ff_out;

void unpack2_ (long int *IEND)

{
/* ------------------- The CC2 Conversion Table------------------------------ */

float c2[] ={ 1057., 1121., 1185., 1249., 1313., 1377., 1441.,
              1505., 1569., 1633., 1697., 1761., 1825., 1889., 1953., 2017.,
              2113., 2241., 2369., 2497., 2625., 2753., 2881., 3009.,
              3137., 3265., 3393., 3521., 3649., 3777., 3905., 4033.,
              4225., 4481., 4737., 4993., 5249., 5505., 5761., 6017.,
              6273., 6529., 6785., 7041., 7297., 7553., 7809., 8065.,
              8449., 8961., 9473., 9985., 10497., 11009., 11521., 12033.,
              12545., 13057., 13569., 14081., 14593., 15105., 15617., 16129.,
              16897., 17921., 18945., 19969., 20993., 22017., 23041., 24065.,
              25089., 26113., 27137., 28161., 29185., 30209., 31233., 32257.,
              33793., 35841., 37889., 39937., 41985., 44033., 46081., 48129.,
              50177., 52225., 54273., 56321., 58369., 60417., 62465., 64513.,
              67584., 71680., 75776., 79872., 83968., 88064., 92160., 96256.,
              4.25  ,  4.75,   5.25,   5.75,   6.25,   6.75,   7.25,   7.75,
              135168.,143360.,151552.,159744.,167936.,176128.,184320.,192512.,
              2.125 ,  2.375,  2.625,  2.875,  3.125,  3.375,  3.625,  3.875,
         128  0.0625, 0.1875, 0.3125, 0.4375, 0.5625, 0.6875, 0.8125, 0.9375,
              1.0625, 1.1875, 1.3125, 1.4375, 1.5625, 1.6875, 1.8125, 1.9375, 143
              -1.0  , -1.0  , -1.0  , -1.0  , -1.0  , -1.0  , -1.0  , -1.0  ,
              8.5   , 9.5   , 10.5  , 11.5  , 12.5  , 13.5  , 14.5  , 15.5 159
              17.   , 18.   , 19.   , 20.   , 21.   , 22.   , 23.   , 24.   ,
              25.   , 26.   , 27.   , 28.   , 29.   , 30.   , 31.   , 32.   ,
         176  34.   , 36.   , 38.   , 40.   , 42.   , 44.   , 46.   , 48.   ,
              50.   , 52.   , 54.   , 56.   , 58.   , 60.   , 62.   ,
              64.   , 67.   , 71.   , 75.   , 79.   , 83.   , 87.   ,
              91.   , 95.   , 99.   , 103.  , 107.  , 111.  , 115.  , 119.  ,
              123.  , 127.  , 133.  , 141.  , 149.  , 157.  , 165.  ,
              173.  , 181.  , 189.  , 197.  , 205.  , 213.  ,
              221.  , 229.  , 237.  , 245.  , 253.  , 265.  , 281.  , 297.  ,
              313.  , 329.  , 345.  , 361.  , 377.  ,
```

```
              393.  , 409.  , 425.  , 441.  , 457.  , 473.  , 489.  , 505.  ,
              529.  , 561.  , 593.  , 625.  ,
              657.  , 689.  , 721.  , 753.  , 785.  , 817.  , 849.  , 881.  ,
              913.  , 945.  , 977.  , 1009.};
/* ------------------ The CCl Conversion Table ---------------------------- */
float ic[]= {1057., 1121., 1185., 1249., 1313., 1377., 1441.,
             1505., 1569., 1633., 1697., 1761., 1825., 1889., 1953.,2017.,
             2113., 2241., 2369., 2497., 2625., 2753., 2881., 3009.,
             3137., 3265., 3393., 3521., 3649., 3777., 3905., 4033.,
             4225., 4481., 4737., 4993., 5249., 5505., 5761., 6017.,
             6273., 6529., 6785., 7041., 7297., 7553., 7809., 8065.,
             8449., 8961., 9473., 9985., 10497., 11009., 11521.,12033.,
            12545., 13057., 13569., 14081., 14593., 15105., 15617.,16129.,
            16897., 17921., 18945., 19969., 20993., 22017., 23041.,24065.,
            25089., 26113., 27137., 28161., 29185., 30209., 31233.,32257.,
            33793., 35841., 37889., 39937., 41985., 44033., 46081.,48129.,
            50177, 52225, 54273, 56321, 58369, 60417, 62465, 64513,
            67585, 71681, 75777, 79873, 83969, 88065, 92161, 96257,
       100353,104449,108545,112641,116737,120833,124929,129025,
       135169,143361,151553,159745,167937,176129,184321,192513,
       200705,208897,217089,225281,233473,241665,249857,258049,
       270337,286721,303105,319489,333873,352257,368641,385025,
       401409,417793,434177,450561,466945,483329,499713,
       0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
       21,22,23,24,25,26,27,28,29,30,31,32,34,36,38,40,42,44,
       46,48,50,52,54,56,58,60,62,64,67,71,75,79,83,87,91,95,
       99,103,107,111,115,119,123,127,133,141,149,157,165,173,
       181,189,197,205,213,221,229,237,245,253,265,281,297,
       313,329,345,361,377,393,409,425,441,457,473,489,505,
       529,561,593,625,657,689,721,753,785,817,849,881,913,
       945,977,1009};
long temp;
short int s_temp;
#if 0
printf (" Size of struct IN_P = %d ",sizeof(struct IN_P));
printf (" --- r_.IOPEN = %d ",r_.IOPEN);
#endif
/* ---------- Read a buffer from the PACKED input file
              and set the returning IEND flag== 1 if EOF
              is detected ------------------------------------ */
/*   printf (" Read a buffer ..\n");  */
     num_char = fread (&in_p_, sizeof_IN_P, 1, ff_in);
     if (num_char != 1) {
        *IEND = 1;
         r_.IEND = 1;
         printf( " EOF encountered.... \n");
         fclose (ff_in);
         return;
     }
/* --------- Unpack IHD[] into their engineering values ---- */
/*   printf(" Start processing a record \n");  */
     for (i=0; i<3; i++) {
        s_temp = r_.IHD[i] = in_p_.IHD[i];
        r_.IHD[i] = ntohs (s_temp);
```

```
      }
   for (i=3; i<8; i++) {
      s_temp = r_.IHD[i+1] = in_p_.IHD[i];
      r_.IHD[i+1] = ntohs (s_temp);
      }
      r_.IHD[3]    = ntohl(in_p_.IHD_Msec);
/* -------- Unpack HEAD[] ---------------------------------- */
   for (i=0; i<2; i++) {
      temp = ntohl (in_p_.JHEAD[i]);
      r_.HEAD[i] = 0.01 * (float) (temp);
      }
   for (i=2; i<6; i++) {
      temp = ntohl (in_p_.JHEAD[i]);
      r_.HEAD[i] = (float) (temp);
      }
      temp = ntohl (in_p_.JHEAD[6]);
      r_.HEAD[6] = 0.01 * (float) (temp);
      temp = ntohl (in_p_.JHEAD[7]);
      r_.HEAD[7] = 0.01 * (float) (temp);
   for (i=8; i<12; i++) {
      temp = ntohl (in_p_.JHEAD[i]);
      r_.HEAD[i] = (float) (temp);
      }
   for (i=12; i<22; i++){
      temp = ntohl (in_p_.JHEAD[i]);
      r_.HEAD[i] = 0.01 * (float) (temp);
      }
      r_.HEAD[22] = (float) (in_p_.Prog_Ver);
/* -------- Unpack ISTAT_P[] ---------------------------- */
   for (i=0; i<2; i++) {
      s_temp = ntohs (in_p_.ISTAT_P[i]);
      r_.ISTAT_P[i] = (long) (s_temp);
      }
r_.ISTAT[0]= (r_.ISTAT_P[0] & 0x80) >> 7;
r_.ISTAT[1]= (r_.ISTAT_P[0] & 0x40) >> 6;
r_.ISTAT[2]= (r_.ISTAT_P[0] & 0x20) >> 5;
r_.ISTAT[3]= (r_.ISTAT_P[0] & 0x10) >> 4;
r_.ISTAT[4]= (r_.ISTAT_P[0] & 0x08) >> 3;
r_.ISTAT[5]= (r_.ISTAT_P[0] & 0x6)  >> 1;
r_.ISTAT[6]= (r_.ISTAT_P[0] & 0x1);
r_.ISTAT[7]=r_.ISTAT_P[1] & 0xFF;
/* -------- Unpack HOUS[] ---------------------------------- */
   for (i=0; i<4; i++) {
      s_temp = ntohs (in_p_.JHOUS[i]);
      r_.HOUS[i] = ((float) (s_temp)) / 10.0;
      }
      s_temp = ntohs (in_p_.JHOUS[4]);
      r_.HOUS[4] = (float) (s_temp);
   for (i=5; i<7; i++) {
      s_temp = ntohs (in_p_.JHOUS[i]);
      r_.HOUS[i] = ((float) (s_temp)) / 10.0;
      }
      s_temp = ntohs (in_p_.JHOUS[7]);
      r_.HOUS[7] = ((float) (s_temp)) / 100.0;
```

```
      for (i=8; i<12; i++) {
         s_temp = ntohs (in_p_.JHOUS[i]);
         r_.HOUS[i] = (float) (s_temp);
         }
         s_temp = ntohs (in_p_.JHOUS[12]);
         r_.HOUS[12] = ((float) (s_temp)) / 100.0;
      for (i=13; i<15; i++) {
         s_temp = ntohs (in_p_.JHOUS[i]);
         r_.HOUS[i] = ((float) (s_temp)) / 10.0;
         }
/* -------- Unpack MEPI[] -------------------------------- */
      for (i=0; i<2; i++)
         r_.MEPI[i] = (int) (ic[ (in_p_.Index_MEPI[i]) ]);

/* -------- Unpack MEP[][] ------------------------------- */
      for (i=0; i<4; i++) {
         for (j=0; j<19; j++) {
               r_.MEP[i][j] = (int) (ic[ (in_p_.Index_MEP[i][j]) ]);
               }
         }
/* -------- Unpack IHEP[][] ------------------------------ */
      for (i=0; i<2; i++) {
         for (j=0; j<11; j++) {
               r_.IHEP[i][j] = (int) (ic[ (in_p_.Index_IHEP[i][j]) ]);
               }
         }
/* -------- Unpack TED[][] ------------------------------- */
      for (i=0; i<4; i++) {
         for (j=0; j<5; j++) {
               r_.TED[i][j] = ic[ (in_p_.Index_TED[i][j]) ];
               }
               r_.TED[i][5] = (float) (in_p_.Index_TED[i][5]);
       for (j=6; j<18; j=j+3) {
               r_.TED[i][j]   = c2[ (in_p_.Index_TED[i][j]) ];
               r_.TED[i][j+1] = ic[ (in_p_.Index_TED[i][j+1]) ];
               r_.TED[i][j+2] = (float) (in_p_.Index_TED[i][j+2]);
               }
/* -------- Unpack TEDFX[] ------------------------------- */
         temp = ntohl (in_p_.JTEDFX[i]);
         r_.TEDFX[i] = ((float) (temp))  / 1000.0;
      }              /* End of for i=0;i<4 loop */
/*  copy data to archive record which users are expecting --- */
      for (i=0;i<9;i++)
               rec_.IHD[i] = r_.IHD[i];
      for (i=0;i<23;i++)
               rec_.HEAD[i] = r_.HEAD[i];
      for (i=0;i<8;i++)
               rec_.ISTAT[i] = r_.ISTAT[i];
      for (i=0;i<15;i++)
               rec_.HOUS[i] = r_.HOUS[i];
      for (i=0;i<2;i++)
               rec_.MEPI[i] = r_.MEPI[i];
      for (i=0;i<4;i++) {
               for (j=0;j<19;j++) {
                        rec_.MEP[i][j] = r_.MEP[i][j];
```

```
                }
        }
        for (i=0;i<2;i++) {
                for (j=0;j<11;j++) {
                        rec_.IHEP[i][j] = r_.IHEP[i][j];
                }
        }
        for (i=0;i<4;i++) {
                for (j=0;j<18;j++) {
                        rec_.TED[i][j] = r_.TED[i][j];
                }
        }
        for (i=0;i<4;i++)
                rec_.TEDFX[i] = r_.TEDFX[i];
/* ---------- END of unpack2.c ------------------------------ */
}
/* ================================================================
   FUNCTION: open_archive.c ()
             'open_archive.c' function is used to open a disk file
             which has been 'dd' from a 3480 packed archive
             data file. 'open_archive.c()' requests user input
             for the packed filename, opens the file for binary
             read.
   ============================================================= */
void open_archive_()
{
       char filename[80];
       printf (" == TIROS / NOAA UNPACK2 Routine ==\n");
       printf (" Enter the PACKED filename (e.g. N72141.NEW) :");
       scanf ("%s",filename);
       if((ff_in = fopen(filename, "rb")) == NULL) {
          printf(" *** Cannot open PACKED input file. STOP !!!");
          fclose(ff_in);
          exit(1);
       };
}
```

A.6. Converting Telemetered Values

The conversion tables are used to convert the telemetry words to counts per accumulation period and are in two arrays called CC1 and CC2. The arrays are used as look-up tables to convert telemetry data words into counts per accumulation period. The CC1 and CC2 arrays each contain 256 32-bit words. Telemetry data words are always one byte (8 bits) in the range 0−255. To convert a telemetry word to counts, use the value of the telemetry word plus 1 (telemetry word + 1, range 1−256) as an index to the CC1 or CC2 array. The number indexed from the array will give counts per accumulation period. To convert counts per accumulation period to counts per second, see the record-timing description and tables in Section 5.

The CC1 array is used for all telemetry data words except the TED total flux channels (0EF−D, 30EF−D, 0PF−D and 30PF−D), which use the CC2 array. Of course, conversion is needed only for values that measure counts. Remember that the TED interval in which the maximum counts are accumulated and the total energy flux values are *not* counts.

The CC2 array is the same as CC1, except for CC2(105) through CC2(160), so it is not necessary to hold both arrays. In addition, CC2(145) through CC2(152) are not referenced by valid telemetry data and so are set to −1 in the array.

Conversion Algorithms

Instead of the CC1 and CC2 conversion tables, telemetry words can be converted to counts per accumulation period by using conversion algorithms. The algorithm variables are as follows:

CTS = counts per accumulation period
Y = the four most significant bits of the telemetry word
X = the four least significant bits of the telemetry word
Note: Use only the integer portion of CTS.

CC1 normal conversion:

If Y = 0 thru 8	and X = 0 through 15,	$CTS = [(X + 16.5) * 2^{(Y+6)}] + 1$
except if Y = 8	and X = 15,	CTS = 0.0
If Y = 9	and X = 0 through 15,	CTS = X + 1.5
If Y = 10	and X = 0 through 15,	CTS = X + 17.5
If Y = 11 thru 15	and X = 0 through 15,	$CTS = [(X + 16.5) * 2^{(Y-10)}] + 1$

CC1 alternate conversion:

Add 113 (01110001 binary) to the word

If Y = 0,	CTS = X
If Y = 1 through 15,	$CTS = [(X + 15.5) * 2^{(Y-1)}] + 1$

Note: If Y > 2 and X = 0, add $2^{(Y-3)}$
If Y = 2 and X = 0, add 1

CC2 conversion for TED total flux channels 0EF−D, 0PF−D, 30EF−D, 30PF−D:

Use the CC1 conversion, with these exceptions:

If Y = 9 and X = 0 through 7, This is an ERROR; this configuration cannot occur

If Y = 9	and X = 8 through 15,	CTS = X + .5
If Y = 8	and X = 0 through 15,	CTS = .125X + .0625
If Y = 7	and X = 0 through 7,	use CC1 except subtract 1
If Y = 7	and X = 8 through 15,	CTS = .25X + .125
If Y = 6	and X = 0 through 7,	use CC1 except subtract 1
If Y = 6	and X = 8 through 15,	CTS = .5X + .25

TED 0DE−M, 0DP−M, 30DE−M and 30DP−M:

Use CC1 conversion, except:

If Y = 0 and X = 0, check corresponding F−D channel

if counts are present in the F−D channel,	CTS = 1057.
if counts are not present in the F−D channel,	CTS = 0.0.

APPENDIX B: Problems and Errors in the Archive Data

Some problems have been found in the SEM data and the archive tapes since processing began in late 1978. The TED instrument data are analyzed carefully when the archive tapes are produced, and all known problems with the data are noted below. The MEPED and HEPAD data have not been checked consistently, and little is known about minor instrument problems. All problems that were not corrected prior to 1993 are described briefly in this section. Problems corrected in the 1993 archive program are described in Appendix C.

The NOAA-7 TED 30° detector "sees" sunlight, which introduces a background response. This response occurs when the satellite is on the day side of Earth and is most apparent in the Northern Hemisphere during Northern-Hemisphere summer and in the Southern Hemisphere during Southern-Hemisphere summer. The undesirable detector response maximizes when the satellite crosses the latitude of the sub-solar point but is still observed to be present well into the auroral regions. The 0° detector is unaffected. When the contaminated 30° detector response is combined with the normal 0° response to obtain the total energy flux at locations below the auroral zones (where the contamination is largest), an anomalous energy flux of about 0.1 mW/m^2 is obtained. Those instances when the 30° detector contamination gives rise to totally incorrect total energy fluxes may be reliably identified using the following criterion: If the satellite is on the day side of Earth and the 30° detector response exceeds the 0° detector response by more than a factor of 3, the 30° detector response should be ignored.

In 1982, NOAA-7 sustained a loss of TED data from day 068, 1145UT, through day 082, 1403UT. During that time the TED instrument was turned off in the course of testing for the source of interference with the NOAA-7 command receiver and then turned on again in an incorrect mode.

In 1982 there was another loss of TED data on NOAA-7 from day 328, 0100UT, through day 394, 1625UT. The TED was again turned off to check the source of contamination being sensed by a "contamination detector" on the spacecraft. After the instrument was turned back on again there was evidence of an electrical interference source or corona, which affected the 30° proton detector in a spasmodic fashion for many days. This problem eventually went away, but while it was present it caused large count rates to appear in the 30° proton channel; these in turn resulted in large and incorrect total energy fluxes.

When NOAA-8 came on line, the "sunlight" problem was more severe than that on NOAA-7. A modification to the archive program removes this contamination.

On April 1, 1982, the MEPED and HEPAD instruments on NOAA-7 were permanently turned off .

There is no HEPAD instrument on NOAA-8, NOAA-10, or NOAA-12, and no HEPAD will be flown on later satellites.

Neither NOAA-9 nor NOAA-11 carried a SEM instrument.

Beginning on NOAA-10 a data-quality flag was calculated during archive processing and inserted into the HEPAD words (which are normally zero because there is no HEPAD instrument after NOAA-7).

IHEP(1,1) = 1 if some of the data in the last 4 seconds of the previous record are questionable.

IHEP(2,1) = 1 if some of the data in the first 4 seconds of this record are questionable.

IHEP (1,2) = 1 if some of the data in the first 4 seconds of this record are questionable.

IHEP(2,2) = 1 if some of the data in the last 4 seconds of this record are questionable.

NOTE: IHEP is the HEPAD array created by the retrieval routines described in Appendix A. On the archive tape these words are actually 143 and 144, so that they are converted to 0 and 1 by the look-up table; where 0 means all data is good, 1 means some data is questionable.

Table B.1 shows the dates and times of minor data problems.

Table B.1. Problems in the SEM data

Spacecraft	Year	Begin DOY/time	End DOY/time	Problem
TIROS-N	1979	188/2025UT	218/1324UT	TED data bad
TIROS-N	1980	067/0130UT	068/0130UT	All SEM data duplicate day-68 observations
TIROS-N	1980	268/1900UT	269/0730UT	Bad orbital data
NOAA-6	1980	062/0430UT	064/0000UT	Bad orbital data
NOAA-6	1980	329/0615UT	329/2300UT	Bad orbital data
NOAA-6	1981	250/1545UT	265/1443UT	No TED data, improper operating mode

APPENDIX C: Corrections to the Archive Program in 1993

Corrections in three areas were introduced in the 1993 version of the archive program: computation of the MEPED pitch angles for the 90° detectors, data quality flagging in the TED data, and angular integration of TED data to obtain omnidirectional energy fluxes in the southern hemisphere.

C.1. MEPED Pitch Angle

Beginning with NOAA–10, the pitch angle of the detector is the same as the pitch angle of the particles it measures. The particle pitch angle is defined as the angle between the particle's velocity angle and the local magnetic field direction. This convention means that a particle precipitating in the northern hemisphere will have a pitch angle less than 90° and a particle precipitating in the southern hemisphere will have a pitch angle greater than 90°.

To determine the direction of the satellite, the subsatellite latitude is checked for two consecutive records (making certain that the records are separated in time by 8000 milleseconds). The satellite is northound if the second latitude is algebraically larger than or equal to the first. In case of a time jump, the current direction is assigned to the last record before the jump.

C.2. Test for Bad TED values

The following tests are implemented in the archive program to further test for bad TED values. The first test determines if the values for the intervals with the maximum counts accumulated for the 0EM, 30 EM, 0PM, and 30 PM are between 1 and 11. This tests the validity of the characteristic energy index. If any of those tests fail, the value of the TEDFX is set to 1000.

Next, the total energy flux readouts are tested to determine if the values are less than zero. If they are, the corresponding value of TEDFX is set to 1000.

The next level of checking involves the magnitudes of the total directional energy flux and their corresponding flux at the maximum interval. If all of these channels have values less than 1000 the data associated with TEDFX can be accepted as valid. If any of these channels have values of more than 1000 the following additional tests should be run on the data:

The first additional test checks for consistency between the total flux measurement and the sensor response in the channel where the maximum response occurred. It assumes that the entire sensor response during a sweep occurred in one channel, and it makes certain that the sweep integrated response is at least that value. If this condition is not true, TEDFX is set to 1000.

The data are then tested to make sure that the sweep-integrated sensor response does not exceed the value that would occur if all 11 channels had the same response as that in the maximum channel. If this value is exceeded, TEDFX is set to 1000.

The TED values are now checked to determine if the values of the directional energy flux and the interval where the maximum occurred are both 1057, or if the directional energy flux is 1121 and the inverval is 1057. If either of these cases is true, TEDFX should be set to 1000.

The final line of defense is to test for the degree of anisotropy between the total flux measured by the 0° and 30° detectors. The ratio between the two responses should not exceed 8 for ions or 100 for electrons.

C.3. Computation of Total Energy Flux

There are two reasons for recomputing the energy fluxes. The first is that the computation was done improperly in the southern hemisphere until 1984. The second is that improved data quality checking has identified cases of bad data not identified before, and, by the same token, has accepted good data that had previously been labeled as questionable. The computation of net energy flux into the atmosphere is done for all instances where the data have been identified as good. Basically, the computation introduces the instrument calibration into the data and performs a crude integration over pitch angle to obtain an estimate of the energy flux into the atmosphere. The calibration used is that for TIROS-N, the only instrument that was well calibrated.

There are three cases to be treated in the angular integration:

1. Both TED detector systems are viewing particles that do not reach the atmosphere, i.e., have magnetically mirrored at altitudes above 120 km. This occurs at low latitude. In such a case the corresponding value of TEDFX is set to 0.0.

2. One, but not both, of the TED look directions are viewing particles that are deposited into the atmosphere. This occurs at low and mid latitudes. In this case the integrated energy flux is calculated, using data from the electron and ion detectors that are viewing the precipitating particle and assuming angular isotropy.

3. Both detector look directions are viewing particles that precipitate into the atmosphere. In this case data from both look directions are used to compute the integrated energy input, and we assume that each look direction measures a flux that is representative of the fluxes over the pitch angle range to a point midway between the two look directions. This occurs at all auroral latitudes. It must be noted that at certain locations the "0 degree" TED detectors actually view a higher pitch angle than do the "30 degree" TED detectors.

*U.S. GOVERNMENT PRINTING OFFICE:1995-673-018/00080