19 January 2010
Tycho

Program PacketA_eH3 is designed to read the latest STEREO * .fin files (or the latest *.ptp files) and produce an ASCII listing of 1-minute flux data for electrons and protons (hydrogen) for the HET telescope.   It makes minor assumptions that the user will use it on a PC (these are enumerated below).

An output data line begins with the date and time followed by flux 1 and its error bar, flux 2 and its error bar, etc. for a total of 14 different fluxes:

AHEAD		BEHIND	
0.7 – 1.4	MeV electrons	0.7 – 1.4	MeV electrons
1.4 – 2.8	MeV electrons	1.4 – 2.8	MeV electrons
2.8 – 4.0	MeV electrons	2.8 – 4.0	MeV electrons
13.6 – 15.1	MeV protons	13.6 – 15.0	MeV protons
15.0 – 17.1	MeV protons	14.9 – 16.9	MeV protons
17.0 – 19.3	MeV protons	16.9 – 19.0	MeV protons
20.8 – 23.8	MeV protons	20.6 – 23.8	MeV protons
23.8 – 26.4	MeV protons	23.7 – 26.5	MeV protons
26.3 – 29.7	MeV protons	26.3 – 29.8	MeV protons
29.6 – 33.4	MeV protons	29.8 – 33.7	MeV protons
33.4 – 35.8	MeV protons	33.6 – 36.1	MeV protons
35.6 – 40.5	MeV protons	36.0 – 40.8	MeV protons
40 – 60	MeV protons	40 – 60	MeV protons
60 – 100	MeV protons	60 – 100	MeV protons

Note that the differences between the energy intervals for the Ahead spacecraft and for the Behind spacecraft are fairly small.

The driver program (PacketA_eH3.c) assumes that the data files are either in C:\StereoAheadData or C:\StereoBehindData. The data files which I use have names like het_behind_2006_241_1_05.fin. PacketA_eH3.c searches for the latest version of these files for each day to be processed and then reads them to find HET A packets, i.e. packets with APID = 590. The user provides a year and a month and the program processes data for all the corresponding days.

The output data files will go to the same directory as the input files. The output data files have names like A06DeceH.1m (the one-minute data files I posted on-line at Caltech have names AeH06Dec.1m … I rearranged the characters in the names so as to be able to distinguish them). There are also output files with extension .hist. These history files report primarily on data gaps in the input data.

The code in the driver includes 3 non-ANSI standard library functions: _strdate(), _strtime(), and _getch(). The first two get the current system date and time, while the 3rd returns a single character with a single key-stroke. I have tried to only use ANSI standard library functions in the remaining code, i.e. in the code which Peter Schroeder will use). Apart from these functions and the path names which include the hard drive as C:, I think that this code should compile and run just as well on a UNIX machine.

The driver assumes that the binary input packet data is stored in an unsigned character buffer (called bufferin) of length 298 characters which is created by the driver. The buffer is assumed to contain a ground receipt header + CCSDS header + packet data. The driver passes a pointer to ProcessAPacket(bufferin), which is the desired stand-alone subroutine. The results are stored in another character buffer which is created by the C-code. IDL code can access this data via the pointer pA_PacketOut.