EDSN Beacon Packet Decoding

1. Introduction

EDSN sends AX.25 packets at 437.10MHz AFSK. There are 2 types of packets: State of Health (SOH) and Science.

Each packet contains ASCII values from 32 to 255. Depending on the TNC being used, the data string received could be preceded by the following set of characters:

KE6QLL>UNDEF,TELEM/1: <<UI>>:

This denotes the sender and recipient of the packet. Ignore this when decoding the packet.

The satellite will typically send either SOH or Science packets at a 60 second periodicity. When the satellite is detumbling, soon after launch, it will transmit only SOH packets every 120 seconds. The beacon will not transmit during space-to-space crosslink communication activities, which occur 4 times in a 25-hour period. Each crosslink activity lasts up to 46 minutes.

2. Beacon Packet Format

Each packet type contains different information, and Tables 1-3 show the breakdown of each packet type.

Packets are divided between header and sensor values. The header contains an ASCII character sequence of “EDSN”, an exclamation mark (!) or double quotation mark (“), and a letter between “A” and “H” that do not need decoding. The remaining values are encoded extended ASCII characters from 32 to 255. There are then 224 possible values per character (Base224). Therefore, the number of possible values per variable is 224^n, where *n* is the length of the variable in bytes.

Each packet starts with a sync character sequence of “EDSN”. This is used to find the start of the packet. The fifth character (! or “) denotes the message type. The sixth character identifies the spacecraft that sent the packet (A to H). The “Science Data” group of the science packet uses a Base224 to Base256 decoding (see Section 2.1). The remaining characters are converted from base 224 to decimal values and scaled applying the following formula:

$$R = \sum_{i=0}^{n-1} 224^{n-1-i}(d\_A_i - 32)\frac{(m - m)}{224^n - 1} + m$$

Where sensor data byte order is big-endian and the index of the first byte is i=0, *dec_ASCII* is the decimal value of the ASCII character, *n* is the size of the variable, *max* is the maximum value of the variable and *min* is the minimum value of the variable.

2.1. SOH packet

SOH packets are 187 bytes long and are sent every 120 seconds. These packets have a constant length and have an exclamation mark (!) after the "EDSN" start word. Below is a packet decoding example.

Example of received RF data (in hex format, with checksum):

45 44 53 4E 21 47 21 33 9E 49 98 3E 24 46 20 20 24 88 20 21 21 20 20 20 20 20 20 20 20 20 20 20 20 20 20 22 20 22 22 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 21 F5 D1 FC F9 90 5E 76 E4 68 EC 61 47 EA A6 C3 43 CF 49 49 DA 21 F6 D0 A5 47 21 24 82 60 A0 7A 86 30 A2 A8 A7 E0 8F F2 90 20 90 5F FF FF 20 20 20 20 2F F2 85 BF A2 F4 A8 3C 8F 9B 8F CC 90 42 9B 7D E1 58 88 35 8F E5 90 4C 90 32 64 20 9E 49 98 36 23 2F 20 51 30 B0 A1 DD 20 20 20 20 53 B7 20 29 20 28 20 20 21 20 20 21 20 20 A6 83 A7 67 A7 67 A7 67 A6 A9 39 39 39 39 38 38 4E 9E 9E 49 A2 2B DC

Example after being decoded and scaled (in CSV format, no checksum)

EDSN,33,G,243,1418251550,934,0,1000,1,1,0,0,0,0,0,2,2,2,0,0,0,0,0,0,0,0,1102205202000,-3543725.6877,-2791998.8419,-5149681.4383,3654.2501,4513.3234,-5012.0578,1104707188257,4,1104703578,-88.5412,165.9923,212.8213,-0.0026906,9.9651e-005,0.012656,255,-255,-255,3570,-91.7666,169.0187,215.2106,-0.02003,-0.010264,0.0068759,25.9955,184.9976,-17.9961,-0.052815,0.08869,0.036871,68,0,1418251542,68.4606,0.22708,16.7676,115.9322,0,0,46.1377,0.29446,0.26174,0,1.1212,0,0,1.1212,0,0,26.9751,28.9284,28.9284,28.9284,27.3239,34.0509,28.6715,28.6715,28.6715,28.6715,27.5247,27.5247,20126,1418253771,8.4519

2.2. Science packet

Science packets are 192 bytes long and are sent every 120 seconds. These packets have a constant length and have a double quotation mark (") after the "EDSN" start word. The science payload used on EDSN outputs binary data, which is then encoded in order to confirm to the Base224 RF packet format. In order to decode the "Science Data" group from a Base224 byte array into the original binary data, follow these steps:

1) Determine the number of 8 byte chunks in the encoded data (Science Data has 176 encoded bytes = 22 chunks)
2) Determine the size of the decoded byte array based on 60 bits per chunk (22 chunks = 165 bytes decoded data)
3) Convert each 8 byte chunk to an integer representation
  a) Take the first byte integer and subtract 32 (call the value x)
  *b) Take the x from step 3a and calculate x*224^n, where n=7*
  c) Repeat step 3a and step 3b for n-1, until n=0
  d) Sum all 8 values obtained from step a to step c
4) Convert the integer from step 3 to a 60 bit binary value (http://en.wikipedia.org/wiki/Binary_number#Decimal)
  a) Take the integer from step 3d and divide it by 2
  b) If the remainder of step 4a is =0, then the least-significant bit (call it yn, where n=59) is 0, otherwise it is 1

c) Take the integer result of step 4a and repeat step 4a and step 4b, for n-1, until n=0
5) Repeat step 3 and step 4 for each chunk (22 total) to obtain decoded data (165 bytes total)

Note that two chunks are needed to produce 15 bytes of decoded (Base256) data, one chunk will not give an integer number of decoded bytes. This procedure is almost identical to Base64 decoding, except that it uses a radix-224 representation, rather than a radix-64. Table 4 illustrates the bit order associated with Base225 to Base256 Science Data decoding.

Example of received RF data (in hex format, with checksum):

45 44 53 4E 22 43 20 33 9E 48 C4 D3 20 3F 94 69 B2 A7 65 55 51 20 20 33 C8 2D 6F B9 91 45 60 2A 34 AF 6C 46 FE 23 50 CC
6B 54 F2 EA F5 20 20 4E 36 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 80 73
8E 21 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 60 F2
32 95 39 53 22 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 C0 ED C3 83 53 F4 C5 27 84 38

Example after being decoded and scaled (in CSV format, no checksum):

EDSN, B_Science, C, 19, 1418211379, 31, 1418211379, 8, 25, 22, 10, 30.4464, 27.3797, 600.7844, 550.0335, 5.018, 7.6922, 3.3097, 6.8769, 76, 7.9656, 10.599, 0, 131, 0, 0, 1600, 17, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,

EDSN,34,C,19,1418211379,31,1418211379,8,25,0x16,10,30.4464,27.3797,600.7844,550.0335,5.018,7.6922,3.3097,6.8769 ,76,7.9656,10.599,0x00,0x83,0,0,1600,17,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,41776

3. Conversion of Voltage, Current and Temperature

Voltage, Current and Temperature values need a second conversion from a scaled value to their Engineering Unit. In Tables 1-3, conversions to Engineering Units are added on the last column. The Control Register and CPU status values in the Science Data are composed of bit level flags, but can be viewed as decimal values for assessing nominal behavior.

4. Checksum

The checksum is a modified version of the Fletcher algorithm. The sums modulo 224 are not computed at each byte in the data. The modulo operation is done for the packet data after the summing process defined in the Fletcher algorithm. Each byte of the 16-bit sum is then offset by (+) 32 and then converted to a decimal in base-2.

Table 1: SOH Packet Description (part 1 of 2)

CONTENT				BINARY FORMAT			SCALED VALUES			ENGINEERING UNITS	
Group	Var Name	Description	Final Unit	Offset	Bytes	Bits	Range Min	Range Max	Resolution	Valid Range	Conversion
Header	start_word	Start of packet defintion, "EDSN" in ASCII	[Characters]	0	4	32	~	~	~	ASCII 'EDSN'	*[Binary to ASCII]*
	msg_type	Message Type Definition (value 33 to 256, defined on "Types" tab)	[Character]	4	1	8	~	~	~	dec 33	*[Binary to ASCII]*
	src_id	Identification Symbol "A"-"Z" of the spacecraft sending the packet, in ASCII	[Character]	5	1	8	~	~	~	ASCII A-H	*[Binary to ASCII]*
	msg_num	Message Number (separate count for each message type, starts with zero, loops after max value)	[Count]	6	2	16	0	50175	1.000		*[Base224 to decimal]*
	time_s	Posix Time message created (4B for s and 2B for ms, from 1970/1/1 00:00)	Seconds	8	4	32	0	2517630975	1.000		*[Base224 to decimal]*
	time_ms	Posix Time message created (4B for s and 2B for ms, from 1970/1/1 00:00, only uses 0 to 999)	mSeconds	12	2	16	0	50175	1.000	dec 0-999	*[Base224 to decimal]*
System	phone_reboots	number of phone reboots (can rail at 50175)	[Count]	14	2	16	0	50175	1.000		*[Base224 to decimal]*
	router_reboots	number of router reboots (initialized at 1000 due to known bug, can rail at 50175)	[Count]	16	2	16	0	50175	1.000		*[Base224 to decimal]*
	wd_reboots	number of WD reboots (can rail at 50175)	[Count]	18	2	16	0	50175	1.000		*[Base224 to decimal]*
	gps_fix	GPS fix counter (first valid GPS vector in activity, loops at 255, remains 223 for 32 extra counts)	[Count]	20	1	8	0	223	1.000		*[Base224 to decimal]*
	is_captain	isCaptain flag ("0"=false or "1"=true)	[Count]	21	1	8	~	~	~	ASCII 0 to 1	*[Binary to ASCII]*
RF Communication	last_dl_start_s	Last Captain Data Structure downlink start time (from 1970/1/1 00:00)	Seconds	22	4	32	0	2517630975	1.000		*[Base224 to decimal]*
	next_dl_start_s	Next Downlink activity start time (from 1970/1/1 00:00, only populated for captain)	Seconds	26	4	32	0	2517630975	1.000		*[Base224 to decimal]*
	dl_lock	Downlink contact count (when first MHX lock signal receieved in activity, updates after event, loops at 255)	[Count]	30	1	8	0	223	1.000		*[Base224 to decimal]*
	dl_tx	Count of total downlink packets sent (when a downlink packet is sent; CPT only, updates after event)	[Count]	31	2	16	0	50175	1.000		*[Base224 to decimal]*
	xl_pkt	Count of total packets created (when a Science or XL SOH packet is created, includes DL science)	[Count]	33	2	16	0	50175	1.000		*[Base224 to decimal]*
	xl_tx	Count of total crosslink packets sent (when a Sci or SOH packet is sent to captain; LT only; no ping)	[Count]	35	2	16	0	50175	1.000		*[Base224 to decimal]*
	xl_sessions	Number of comm sessions (counts first ping received for XL session, updates after event; LT only, loops at 255)	[Count]	37	1	8	0	223	1.000		*[Base224 to decimal]*
	xl_Rx	Count of total crosslink packets received (when a CPT receives a packet from a crosslink target LT)	[Count]	38	2	16	0	50175	1.000		*[Base224 to decimal]*
	cross_rx_A	Total valid packets received from SC-A (when a CPT receives a packet from LT SC-A, no pings)	[Count]	40	2	16	0	50175	1.000		*[Base224 to decimal]*
	cross_rx_B	Total valid packets received from SC-B (when a CPT receives a packet from LT SC-B, no pings)	[Count]	42	2	16	0	50175	1.000		*[Base224 to decimal]*
	cross_rx_C	Total valid packets received from SC-C (when a CPT receives a packet from LT SC-C, no pings)	[Count]	44	2	16	0	50175	1.000		*[Base224 to decimal]*
	cross_rx_D	Total valid packets received from SC-D (when a CPT receives a packet from LT SC-D, no pings)	[Count]	46	2	16	0	50175	1.000		*[Base224 to decimal]*
	cross_rx_E	Total valid packets received from SC-E (when a CPT receives a packet from LT SC-E, no pings)	[Count]	48	2	16	0	50175	1.000		*[Base224 to decimal]*
	cross_rx_F	Total valid packets received from SC-F (when a CPT receives a packet from LT SC-F, no pings)	[Count]	50	2	16	0	50175	1.000		*[Base224 to decimal]*
	cross_rx_G	Total valid packets received from SC-G (when a CPT receives a packet from LT SC-G, no pings)	[Count]	52	2	16	0	50175	1.000		*[Base224 to decimal]*
	cross_rx_H	Total valid packets received from SC-H (when a CPT receives a packet from LT SC-H, no pings)	[Count]	54	2	16	0	50175	1.000		*[Base224 to decimal]*
GPS (only updates when GPS locks)	gps_time	GPS time in milliseconds (from 1980/1/6 00:00; accounts for 16 leap seconds)	mSeconds	56	6	48	0	1.26325E+14	1.000		*[Base224 to decimal]*
	gps_pos_x	Most recent GPS position X: ref to center of earth (ECEF)	meters	62	3	24	-8000000	8000000	1.424		*[Base224 to decimal & scaled]*
	gps_pos_y	Most recent GPS position Y: ref to center of earth (ECEF)	meters	65	3	24	-8000000	8000000	1.424		*[Base224 to decimal & scaled]*
	gps_pos_z	Most recent GPS position Z: ref to center of earth (ECEF)	meters	68	3	24	-8000000	8000000	1.424		*[Base224 to decimal & scaled]*
	gps_vel_x	Most recent GPS velocity X (ECEF)	meters/sec	71	2	16	-8000	8000	0.319		*[Base224 to decimal & scaled]*
	gps_vel_y	Most recent GPS velocity Y (ECEF)	meters/sec	73	2	16	-8000	8000	0.319		*[Base224 to decimal & scaled]*
	gps_vel_z	Most recent GPS velocity Z (ECEF)	meters/sec	75	2	16	-8000	8000	0.319		*[Base224 to decimal & scaled]*
	gps_posix_ms	Posix time at moment when GPS time is received (from 1970/1/1 00:00; accounts for 16 leap seconds)	mSeconds	77	6	48	0	1.26325E+14	1.000		*[Base224 to decimal]*
ADCS (only updates when alignment/detumble activity runs)	ACS Mode	Recently completed ACS activity (bdot data only updates for Detumble and MagAlign, see "Types" tab)	[Character]	83	1	8	~	~	~	ASCII 1-4	*[Binary to ASCII]*
	bdot_time	(1) start time of Detumble/MagAlign (from 1970/1/1 00:00)	Seconds	84	4	32	0	2517630975	1.000		*[Base224 to decimal]*
	bdot_mag_x	(1) Non-calibrated magnetic field intensity-x	uTesla	88	2	16	-999	999	0.040		*[Base224 to decimal & scaled]*
	bdot_mag_y	(1) Non-calibrated magnetic field intensity-y	uTesla	90	2	16	-999	999	0.040		*[Base224 to decimal & scaled]*
	bdot_mag_z	(1) Non-calibrated magnetic field intensity-z	uTesla	92	2	16	-999	999	0.040		*[Base224 to decimal & scaled]*
	bdot_gyro_x	(1) calibrated spin rate-x (should be greater than final to verify maneuver phase performance)	Radians/sec	94	2	16	-5	5	0.000		*[Base224 to decimal & scaled]*
	bdot_gyro_y	(1) calibrated spin rate-y (should be greater than final to verify maneuver phase performance)	Radians/sec	96	2	16	-5	5	0.000		*[Base224 to decimal & scaled]*
	bdot_gyro_z	(1) calibrated spin rate-z (should be greater than final to verify maneuver phase performance)	Radians/sec	98	2	16	-5	5	0.000		*[Base224 to decimal & scaled]*
	bdot_magtor_x	(1) magnetorquer value-x (typically railed)	[Cmd value]	100	2	16	-255	255	0.010		*[Base224 to decimal & scaled]*
	bdot_magtor_y	(1) magnetorquer value-y (typically railed; high during alignment)	[Cmd value]	102	2	16	-255	255	0.010		*[Base224 to decimal & scaled]*
	bdot_magtor_z	(1) magnetorquer value-z (typically railed)	[Cmd value]	104	2	16	-255	255	0.010		*[Base224 to decimal & scaled]*
	bdot_dtime	(C) time since start of Detumble/MagAlign	Seconds	106	2	16	0	50175	1.000		*[Base224 to decimal]*
	bdot_mag_x	(C) Non-calibrated magnetic field intensity-x	uTesla	108	2	16	-999	999	0.040		*[Base224 to decimal & scaled]*
	bdot_mag_y	(C) Non-calibrated magnetic field intensity-y	uTesla	110	2	16	-999	999	0.040		*[Base224 to decimal & scaled]*
	bdot_mag_z	(C) Non-calibrated magnetic field intensity-z	uTesla	112	2	16	-999	999	0.040		*[Base224 to decimal & scaled]*
	bdot_gyro_x	(C) calibrated spin rate-x (should be <2deg/s for final detumble, <1deg/s for alignment)	Radians/sec	114	2	16	-5	5	0.000		*[Base224 to decimal & scaled]*
	bdot_gyro_y	(C) calibrated spin rate-y (should be <2deg/s for final detumble, <1deg/s for alignment)	Radians/sec	116	2	16	-5	5	0.000		*[Base224 to decimal & scaled]*
	bdot_gyro_z	(C) calibrated spin rate-z (should be <2deg/s for final detumble, <1deg/s for alignment)	Radians/sec	118	2	16	-5	5	0.000		*[Base224 to decimal & scaled]*
	bdot_magtor_x	(C) magnetorquer value-x (typically railed)	[Cmd value]	120	2	16	-255	255	0.010		*[Base224 to decimal & scaled]*
	bdot_magtor_y	(C) magnetorquer value-y (typically railed; high during alignment)	[Cmd value]	122	2	16	-255	255	0.010		*[Base224 to decimal & scaled]*
	bdot_magtor_z	(C) magnetorquer value-z (typically railed)	[Cmd value]	124	2	16	-255	255	0.010		*[Base224 to decimal & scaled]*
	bdot_bdot_X	(C) Bdot X	uTesla/Sec	126	2	16	-50	50	1.993E-03		*[Base224 to decimal & scaled]*
	bdot_bdot_Y	(C) Bdot Y	uTesla/Sec	128	2	16	-50	50	1.993E-03		*[Base224 to decimal & scaled]*
	bdot_bdot_Z	(C) Bdot Z	uTesla/Sec	130	2	16	-50	50	1.993E-03		*[Base224 to decimal & scaled]*
	Alignment Error	Average measured magnetic field to target magnetic field vector, absolute angle (non-calibrated)	Radians	132	1	8	0	3.2	0.014		*[Base224 to decimal & scaled]*
	Pointing Error	Average measured sun to target sun vector absolute angle (T0+160 to T0+300 in Pointing Packet)	Radians	133	1	8	0	3.2	0.014		*[Base224 to decimal & scaled]*

Table 2: SOH Packet Description (part 2 of 2)

CONTENT				BINARY FORMAT			SCALED VALUES			ENGINEERING UNITS	
Group	Var Name	Description	Final Unit	Offset	Bytes	Bits	Range Min	Range Max	Resolution	Valid Range	Conversion
EPS Current	SI_time	Posix time of Sensor Interface readings (EPS Current and Temp, from 1970/1/1 00:00)	Seconds	134	4	32	0	2517630975	1.000		*[Base224 to decimal]*
	i_sat	Current of satellite (original value is an integer)	mAmps	138	2	16	0	1023	0.020	dec 0-1023	mA=4.8876*[Base224 to dec & scale]
	i_sten	Current of stensat board (original value is an integer, can saturate)	mAmps	140	2	16	0	1023	0.020	dec 0-1023	mA=0.2273*[Base224 to dec & scale]
	i_EPS	Current of EPS board (original value is an integer)	mAmps	142	2	16	0	1023	0.020	dec 0-1023	mA=0.2206*[Base224 to dec & scale]
	i_phone	Current of phone board (original value is an integer, can saturate)	mAmps	144	2	16	0	1023	0.020	dec 0-1023	mA=0.1955*[Base224 to dec & scale]
	i_ADCS	Current of ADCS Arduino (original value is an integer, can saturate)	mAmps	146	2	16	0	1023	0.020	dec 0-1023	mA=0.2506*[Base224 to dec & scale]
	i_MHX	Current of MHX radio (original value is an integer)	mAmps	148	2	16	0	1023	0.020	dec 0-1023	mA=2.4438*[Base224 to dec & scale]
	i_router	Current of router board (original value is an integer)	mAmps	150	2	16	0	1023	0.020	dec 0-1023	mA=0.1955*[Base224 to dec & scale]
	i_GPS	Current of GPS Receiver (original value is an integer, 3200=0)	mAmps	152	2	16	0	32000	0.638	dec 0-1023	mA=0.0513*[Base224 to dec & scale]
	i_PL	Current of EPISEM Payload (original value is an integer, 3200=0)	mAmps	154	2	16	0	32000	0.638	dec 0-1023	mA=0.0513*[Base224 to dec & scale]
	i_Lithium	Current of Lithium Radio (original value is an integer)	mAmps	156	2	16	0	1023	0.020	dec 0-1023	mA=1.4375*[Base224 to dec & scale]
	i_solarXp	Current of solar panel X+ (original value is an integer)	mAmps	158	1	8	0	1023	4.587	dec 0-1023	r = [Base224 to decimal & scaled] mA=0.2444 * r
	i_solarXn	Current of solar panel X- (original value is an integer)	mAmps	159	1	8	0	1023	4.587	dec 0-1023	
	i_solarYp	Current of solar panel Y+(original value is an integer)	mAmps	160	1	8	0	1023	4.587	dec 0-1023	
	i_solarYn	Current of solar panel Y- (original value is an integer)	mAmps	161	1	8	0	1023	4.587	dec 0-1023	
	i_solarZp	Current of solar panel Z+ (original value is an integer)	mAmps	162	1	8	0	1023	4.587	dec 0-1023	
	i_solarZn	Current of solar panel Z- (original value is an integer)	mAmps	163	1	8	0	1023	4.587	dec 0-1023	
Temperature	t_Lithium	Temp of Lithium Radio (original value is an integer)	Celsius	164	2	16	0	1023	0.020	dec 0-1023	r = [Base224 to decimal & scaled] C = 0.4888 * r - 273.15
	t_EPS	Temp of EPS board (original value is an integer)	Celsius	166	2	16	0	1023	0.020	dec 0-1023	
	t_ADCS_MHX	Temp of ADCS_MHX board (original value is an integer)	Celsius	168	2	16	0	1023	0.020	dec 0-1023	
	t_router	Temp of router board (original value is an integer)	Celsius	170	2	16	0	1023	0.020	dec 0-1023	
	t_sten	Temp of stensat board (original value is an integer)	Celsius	172	1	8	0	1023	4.587	dec 0-1023	
	t_phone	Temp of phone board (original value is an integer)	Celsius	173	1	8	0	1023	4.587	dec 0-1023	
	t_solarXp	Temp of solar panel X+ (original value is an integer)	Celsius	174	1	8	0	1023	4.587	dec 0-1023	r = [Base224 to decimal & scaled] if <512, C = 0.25 * r, if ≥512 C = -0.25*(r - 1024)
	t_solarXn	Temp of solar panel X- (original value is an integer)	Celsius	175	1	8	0	1023	4.587	dec 0-1023	
	t_solarYp	Temp of solar panel Y+ (original value is an integer)	Celsius	176	1	8	0	1023	4.587	dec 0-1023	
	t_solarYn	Temp of solar panel Y- (original value is an integer)	Celsius	177	1	8	0	1023	4.587	dec 0-1023	
	t_solarZp	Temp of solar panel Z+ (original value is an integer)	Celsius	178	1	8	0	1023	4.587	dec 0-1023	
	t_solarZn	Temp of solar panel Z- (original value is an integer)	Celsius	179	1	8	0	1023	4.587	dec 0-1023	
Check	CHKSUM	Checksum for the packet (Modified Fletcher-16, IETF RFC1145; offset each byte by +32, sum, apply modulo 224)	[Check]	180	2	16	"	"	"		
Dyn.	WD_time_s	Posix time of battery voltage measure (updated periodically by WD; before GPS, WD uses run-time)	Seconds	182	4	32	0	2517630975	1.000		*[Base224 to decimal]*
	WD_voltage	Scaled battery voltage (updated periodically by WD, integer)	Volts	186	1	8	0	1023	4.567	dec 0-1023	V=[Base224 to dec & scale]/102.4

Table 3: Science Packet Description

CONTENT				BINARY FORMAT			SCALED VALUES			ENGINEERING UNITS	
Group	Var Name	Description	Final Unit	Offset	Bytes	Bits	Range Min	Range Max	Resolution	Valid Range	Conversion
Header	start_word	Start of packet defintion, "EDSN" in ASCII	[tag]	0	4	32	~	~	~	ASCII "EDSN"	*[Binary to ASCII]*
	msg_type	Message Type Definition (value 33 to 255, defined on "Types" tab)	[character]	4	1	8	~	~	~	dec 34	*[Binary to ASCII]*
	src_id	Identification Symbol "A"-"Z" of the spacecraft sending the packet	[character]	5	1	8	~	~	~	ASCII A-H	*[Binary to ASCII]*
	msg_num	Message Number (separate count for each message type, starts with zero, loops after max value)	[count]	6	2	16	0	50175	1.000		*[Base224 to decimal]*
	time_s	Posix Time message created (4B for s and 2B for ms, from 1970/1/1 00:00)	sec	8	4	32	0	2517630975	1.000		*[Base224 to decimal]*
	time_ms	Posix Time message created (4B for s and 2B for ms, from 1970/1/1 00:00, only uses 0 to 999)	msec	12	2	16	0	50175	1.000	dec 0-999	*[Base224 to decimal]*
Science Data (Entire block is encoded with Base224 algorithm. See instructions below.)	*[Encoding]*	*Base224 encoding overhead (176B encoded for 165B decoded; 8/7.5 ratio)*	*[Encoding]*	14	*11*	*88*	~	~	~		
	pl_start_s	Receipt Time of EPISEM Measurement (4B for s and 1B for ms, from 1970/1/1 00:00)	sec		4	32	0	4294967295	1.000		*[Base224 to Base256 to decimal]*
	pl_start_ms	Receipt Time of EPISEM Measurement (4B for s and 1B for ms, from 1970/1/1 00:00)	msec		1	8	0	999	3.918		*[Base224 to Base256 to dec & scale]*
	pl_data0	Serial Number	[count]		1	8	0	255	1.000		*[Base224 to Base256 to decimal]*
	pl_data1	Control Register (see comment, should return 0x16 after proper command)	[binary]		1	8	~	~	~	hex 16	*[Base224 to Base256 binary]*
	pl_data2	Packet Counter (counter resets at payload restart)	[count]		2	16	0	65535	1.000	dec 0-10	*[Base224 to Base256 to decimal]*
	pl_data4	Temp Monitor 0 (AD590 sensor, +X/+Z corner)	Celcius		1	8	0	255	1.000	dec >0	**r = [Base224 to Base256 to decimal]**
	pl_data5	Temp Monitor 1 (AD590 sensor, -X/-Z corner)	Celcius		1	8	0	255	1.000	dec >0	celcius= 3.06663*r-273.15
	pl_data6	HVPS Volt Mon	Volts		2	16	0	65535	1.000	dec >0	volts=(-1E-4)*r^2+0.82*r-1.75
	pl_data8	HVPS Set Volt	Volts		1	8	0	255	1.000	dec >0	volts=(-2.8898E-4)*r^2+3.1335*r+25.69
	pl_data9	5V Voltage	Volts		1	8	0	255	1.000	dec >0	volts=0.021353*r
	pl_data10	5V Current	mAmps		2	16	0	65535	1.000	dec >0	milliamp=0.035448*r
	pl_data12	3.3V Voltage	Volts		1	8	0	255	1.000	dec >0	volts=0.021353*r
	pl_data13	3.3V Current	mAmps		2	16	0	65535	1.000	dec >0	milliamp=0.035448*r
	pl_data15	FSW Revision (should be 0x4c)	[count]		1	8	0	255	1.000	dec 76	*[Base224 to Base256 to decimal]*
	pl_data16	VBATT Voltage (volts=0.054935*r)	Volts		1	8	0	255	1.000	6.5V-8.5V	volts=0.054935*r
	pl_data17	VBATT Current	mAmps		2	16	0	65535	1.000		milliamp=0.035448*r
	pl_data19	CPU Status (binary, see comment)	[binary]		1	8	~	~	~	hex 00	*[Base224 to Base256 binary]*
	pl_data20	CPU Status (binary, see comment)	[binary]		1	8	~	~	~	hex 83	*[Base224 to Base256 binary]*
	pl_data21	CRC Fail Counter (number of commands received with bad CRCs)	[count]		1	8	0	255	1.000		*[Base224 to Base256 to decimal]*
	pl_data22	Invalid Command Counter	[count]		1	8	0	255	1.000		*[Base224 to Base256 to decimal]*
	pl_data23	Bytes Sent	[count]		3	24	0	16777215	1.000		*[Base224 to Base256 to decimal]*
	pl_data27	Bytes Received (should return 0x11 after proper command)	[count]		2	16	0	65535	1.000	dec 17	*[Base224 to Base256 to decimal]*
	pl_data28	Low Voltage Reset Flag	[count]		1	8	0	255	1.000		*[Base224 to Base256 to decimal]*
	pl_data29	Science Data 1 – 60 (2 bytes per one-second bin)	[count]		120	960	0	65535	1.000		*[Base224 to Base256 to decimal]*
	pl_data149	Spare Data Fields	[none]		9	72	~	~	~		*[Base224 to Base256]*
	pl_data158	CCITT 16-bit CRC of bytes 0 to 157	[crc]		2	16	~	~	~		*[Base224 to Base256]*
Check	CHKSUM	Checksum for the packet (Modified 8-bit Fletcher, IETF RFC1145; offset each byte by +32, sum, apply modulo 224)	[check sum]	190	2	16	~	~	~		

Table 4: Science Data Decoding Byte Order

encoded byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	... 175
encoded bits	0...7	8...15	16...23	24...31	32...39	40...47	48...55	56...63	0...7	8...15	16...23	24...31	32...39	40...47	48...55	56...63	...
chunk number	0								1								... 22
decoded bits	0...7	8...15	16...23	24...31	32...39	40...47	48...55	56...59	0...3	4...11	12...19	20...27	28...35	36...43	44...51	52...59	
decoded byte	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14		... 164