AltOS Telemetry

Packet Definitions

Keith Packard

Revision History

Revision 0.1	01 July 2011
	Initial content

Table of Contents

Packet Format Design

AltOS telemetry data is split into multiple different packets, all the same size, but each includs an identifier so that the ground station can distinguish among different types. A single flight board will transmit multiple packet types, each type on a different schedule. The ground software need look for only a single packet size, and then decode the information within the packet and merge data from multiple packets to construct the full flight computer state.

Each AltOS packet is 32 bytes long. This size was chosen based on the known telemetry data requirements. The power of two size allows them to be stored easily in flash memory without having them split across blocks or leaving gaps at the end.

All packet types start with a five byte header which encodes the device serial number, device clock value and the packet type. The remaining 27 bytes encode type-specific data.

Packet Formats

This section first defines the packet header common to all packets and then the per-packet data layout.

Packet Header

Table 1. Telemetry Packet Header

Off-set	Data Type	Name	Description
0	uint16_t	serial	Device serial Number
2	uint16_t	tick	Device time in 100ths of a second
4	uint8_t	type	Packet type
5			

Each packet starts with these five bytes which serve to identify which device has transmitted the packet, when it was transmitted and what the rest of the packet contains.

Sensor Data

Type	Description
0x01	TeleMetrum Sensor Data
0x02	TeleMini Sensor Data
0x03	TeleNano Sensor Data

TeleMetrum, TeleMini and TeleNano share this same packet format for sensor data. Each uses a distinct packet type so that the receiver knows which data values are valid and which are undefined.

Sensor Data packets are transmitted once per second on the ground, 10 times per second during ascent and once per second during descent and landing

Table 2. Sensor Packet Contents

Off-set	Data Type	Name	Description
5	uint8_t	state	Flight state

Off-set	Data Type	Name	Description
6	int16_t	accel	accelerometer (TM only)
8	int16_t	pres	pressure sensor
10	int16_t	temp	temperature sensor
12	int16_t	v_batt	battery voltage
14	int16_t	sense_d	drogue continuity sense (TM/Tm)
16	int16_t	sense_m	main continuity sense (TM/Tm)
18	int16_t	acceleration	$m/s^2 * 16$
20	int16_t	speed	m/s * 16
22	int16_t	height	m
24	int16_t	ground_pres	Average barometer reading on ground
26	int16_t	ground_accel	TM
28	int16_t	accel_plus_g	TM
30	int16_t	accel_minus_g	TM
32			

Configuration Data

Type	Description
0x04	Configuration Data

This provides a description of the software installed on the flight computer as well as any user-specified configuration data.

Configuration data packets are transmitted once per second during all phases of the flight

Table 3. Sensor Packet Contents

Off-set	Data Type	Name	Description
5	uint8_t	type	Device type
6	uint16_t	flight	Flight number
8	uint8_t	config_major	Config major version
9	uint8_t	config_minor	Config minor version
10	uint16_t	apogee_delay	Apogee deploy delay in seconds
12	uint16_t	main_deploy	Main deploy alt in meters
14	uint16_t	flight_log_max	Maximum flight log size (kB)
16	char	callsign[8]	Radio operator identifier
24	char	version[8]	Software version identifier
32			

GPS Location

Type	Description
0x05	GPS Location

This packet provides all of the information available from the Venus SkyTraq GPS receiver—position, time, speed and precision estimates.

GPS Location packets are transmitted once per second during all phases of the flight

Table 4. GPS Location Packet Contents

Off-set	Data Type	Name	Description
5	uint8_t	flags	See GPS Flags table below
6	int16_t	altitude	m
8	int32_t	latitude	degrees * 10^7
12	int32_t	longitude	degrees * 10^7
16	uint8_t	year	
17	uint8_t	month	
18	uint8_t	day	
19	uint8_t	hour	
20	uint8_t	minute	
21	uint8_t	second	
22	uint8_t	pdop	* 5
23	uint8_t	hdop	* 5
24	uint8_t	vdop	* 5
25	uint8_t	mode	See GPS Mode table below
26	uint16_t	ground_speed	cm/s
28	int16_t	climb_rate	cm/s
30	uint8_t	course	/ 2
31	uint8_t	unused[1]	
32			

Packed into a one byte field are status flags and the count of satellites used to compute the position fix. Note that this number may be lower than the number of satellites being tracked; the receiver will not use information from satellites with weak signals or which are close enough to the horizon to have significantly degraded position accuracy.

Table 5. GPS Flags

Bits	Name	Description
0-3	nsats	Number of satellites in solution
4	valid	GPS solution is valid

Bits	Name	Description
5	running	GPS receiver is operational
6	date_valid	Reported date is valid
7	course_valid	ground speed, course and climb rates are valid

Here are all of the valid GPS operational modes. Altus Metrum products will only ever report 'N' (not valid), 'A' (Autonomous) modes or 'E' (Estimated). The remaining modes are either testing modes or require additional data.

Table 6. GPS Mode

Mode	Name	Decsription
N	Not Valid	All data are invalid
A	Autonomous mode	Data are derived from satellite data
D	Differential Mode	Data are augmented with differential data from a known ground station. The SkyTraq unit in TeleMetrum does not support this mode
E	Estimated	Data are estimated using dead reckoning from the last known data
M	Manual	Data were entered manually
S	Simulated	GPS receiver testing mode

GPS Satellite Data

Type	Description
0x06	GPS Satellite Data

This packet provides space vehicle identifiers and signal quality information in the form of a C/N1 number for up to 12 satellites. The order of the svids is not specified.

GPS Satellite data are transmitted once per second during all phases of the flight.

Table 7. GPS Satellite Data Contents

Offset	Data Type	Name	Description
5	uint8_t	channels	Number of reported satellite information
6	sat_info_t	sats[12]	See Per-Satellite data table below
30	uint8_t	unused[2]	
32			

Table 8. GPS Per-Satellite data (sat_info_t)

Offset	Data Type	Name	Description
0	uint8_t	svid	Space Vehicle Identifier

Off-set	Data Type	Name	Description
1	uint8_t	c_n_1	C/N1 signal quality indicator
2			

Data Transmission

Altus Metrum devices use the Texas Instruments CC1111 microcontroller which includes an integrated sub-GHz digital transceiver. This transceiver is used to both transmit and receive the telemetry packets. This section discusses what modulation scheme is used and how this device is configured.

Modulation Scheme

Texas Instruments provides a tool for computing modulation parameters given a desired modulation format and basic bit rate. For AltOS, the basic bit rate was specified as 38 kBaud, resulting in the following signal parmeters:

Table 9. Modulation Scheme

Parameter	Value	Description
Modulation	GFSK	Gaussian Frequency Shift Keying
Deviation	20.507812 kHz	Frequency modulation
Data rate	38.360596 kBaud	Raw bit rate
RX Filter Bandwidth	93.75 kHz	Receiver Band pass filter bandwidth
IF Frequency	140.62 kHz	Receiver intermediate frequency

Error Correction

The cc1111 provides forward error correction in hardware, which AltOS uses to improve reception of weak signals. The overall effect of this is to halve the available bandwidth for data from 38 kBaud to 19 kBaud.

Table 10. Error Correction

Parameter	Value	Description
Error Correction	Convolutional coding	1/2 rate, constraint length m=4
Interleaving	4 x 4	Reduce effect of noise burst
Data Whitening	XOR with 9-bit PNR	Rotate right with bit 8 = bit 0 xor bit 5, initial value 111111111

TeleDongle packet format

TeleDongle does not do any interpretation of the packet data, instead it is configured to receive packets of a specified length (32 bytes in this case). For each received packet, TeleDongle produces a single line of text. This line starts with the string "TELEM " and is followed by a list of hexadecimal encoded bytes.

```
TELEM 224f01080b05765e00701f1a1bbeb8d7b60b070605140c0006000000000000000003fa988
```

The hexadecimal encoded string of bytes contains a length byte, the packet data, two bytes added by the cc1111 radio receiver hardware and finally a checksum so that the host software can validate that the line was transmitted without any errors.

Table 11. Packet Format

Offset	Name	Example	Description
0	length	22	Total length of data bytes in the line. Note that this includes the added RSSI and status bytes
1 ⋯ length-3	packet	4f ⋯ 00	Bytes of actual packet data
length-2	rssi	3f	Received signal strength. dBm = rssi / 2 - 74
length-1	lqi	a9	Link Quality Indicator and CRC status. Bit 7 is set when the CRC is correct
length	checksum	88	(0x5a + sum(bytes 1 ⋯ length-1)) % 256

History and Motivation

The original AltoOS telemetry mechanism encoded everything available piece of information on the TeleMetrum hardware into a single unified packet. Initially, the packets contained very little data—some raw sensor readings along with the current GPS coordinates when a GPS receiver was connected. Over time, the amount of data grew to include sensor calibration data, GPS satellite information and a host of internal state information designed to help diagnose flight failures in case of a loss of the on-board flight data.

Because every packet contained all of the data, packets were huge—95 bytes long. Much of the information was also specific to the TeleMetrum hardware. With the introduction of the TeleMini flight computer, most of the data contained in the telemetry packets was unavailable. Initially, a shorter, but still comprehensive packet was implemented. This required that the ground station be pre-configured as to which kind of packet to expect.

The development of several companion boards also made the shortcomings evident—each companion board would want to include telemetry data in the radio link; with the original design, the packet would have to hold the new data as well, requiring additional TeleMetrum and ground station changes.