# AltOS Telemetry

## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>License</td>
<td>1</td>
</tr>
<tr>
<td>1. Packet Format Design</td>
<td>1</td>
</tr>
<tr>
<td>2. Packet Formats</td>
<td>2</td>
</tr>
<tr>
<td>2.1. Packet Header</td>
<td>2</td>
</tr>
<tr>
<td>2.2. TeleMetrum v1.x, TeleMini v1.0 and TeleNano Sensor Data</td>
<td>2</td>
</tr>
<tr>
<td>2.3. TeleMega Sensor Data</td>
<td>3</td>
</tr>
<tr>
<td>2.4. TeleMetrum v2 and newer Sensor Data</td>
<td>4</td>
</tr>
<tr>
<td>2.5. TeleMini v3.0 Sensor Data</td>
<td>5</td>
</tr>
<tr>
<td>2.6. Configuration Data</td>
<td>6</td>
</tr>
<tr>
<td>2.7. GPS Location</td>
<td>6</td>
</tr>
<tr>
<td>2.8. GPS Satellite Data</td>
<td>8</td>
</tr>
<tr>
<td>2.9. Companion Data</td>
<td>8</td>
</tr>
<tr>
<td>3. Data Transmission</td>
<td>9</td>
</tr>
<tr>
<td>3.1. Modulation Scheme</td>
<td>9</td>
</tr>
<tr>
<td>3.2. Error Correction</td>
<td>10</td>
</tr>
<tr>
<td>4. TeleDongle serial packet format</td>
<td>10</td>
</tr>
<tr>
<td>5. History and Motivation</td>
<td>11</td>
</tr>
</tbody>
</table>

## License

Copyright © 2021 Bdale Garbee and Keith Packard

This document is released under the terms of the [Creative Commons ShareAlike 3.0 License](https://creativecommons.org/licenses/by-sa/3.0/)

## 1. Packet Format Design

AltOS telemetry data is split into multiple different packets, all the same size, but each includes 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 needs to 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.
2. Packet Formats

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

2.1. Packet Header

Table 1. Telemetry Packet Header

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>uint16_t</td>
<td>serial</td>
<td>Device serial Number</td>
</tr>
<tr>
<td>2</td>
<td>uint16_t</td>
<td>tick</td>
<td>Device time in 100ths of a second</td>
</tr>
<tr>
<td>4</td>
<td>uint8_t</td>
<td>type</td>
<td>Packet type</td>
</tr>
</tbody>
</table>

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.

2.2. TeleMetrum v1.x, TeleMini v1.0 and TeleNano Sensor Data

Table 2. Sensor Packet Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>TeleMetrum v1.x Sensor Data</td>
</tr>
<tr>
<td>0x02</td>
<td>TeleMini v1.0 Sensor Data</td>
</tr>
<tr>
<td>0x03</td>
<td>TeleNano Sensor Data</td>
</tr>
</tbody>
</table>

TeleMetrum v1.x, TeleMini v1.0 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 3. Sensor Packet Contents

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>uint8_t</td>
<td>state</td>
<td>Flight state</td>
</tr>
<tr>
<td>6</td>
<td>int16_t</td>
<td>accel</td>
<td>accelerometer (TM only)</td>
</tr>
<tr>
<td>8</td>
<td>int16_t</td>
<td>pres</td>
<td>pressure sensor</td>
</tr>
<tr>
<td>10</td>
<td>int16_t</td>
<td>temp</td>
<td>temperature sensor</td>
</tr>
<tr>
<td>12</td>
<td>int16_t</td>
<td>v_batt</td>
<td>battery voltage</td>
</tr>
<tr>
<td>14</td>
<td>int16_t</td>
<td>sense_d</td>
<td>drogue continuity sense (TM/Tm)</td>
</tr>
<tr>
<td>16</td>
<td>int16_t</td>
<td>sense_m</td>
<td>main continuity sense (TM/Tm)</td>
</tr>
<tr>
<td>18</td>
<td>int16_t</td>
<td>acceleration</td>
<td>m/s² * 16</td>
</tr>
</tbody>
</table>
### 2.3. TeleMega Sensor Data

**Table 4. TeleMega Packet Type**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x08</td>
<td>TeleMega IMU Sensor Data</td>
</tr>
<tr>
<td>0x09</td>
<td>TeleMega Kalman and Voltage Data</td>
</tr>
</tbody>
</table>

TeleMega has a lot of sensors, and so it splits the sensor data into two packets. The raw IMU data are sent more often; the voltage values don't change very fast, and the Kalman values can be reconstructed from the IMU data.

IMU 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.

Kalman and Voltage Data packets are transmitted once per second on the ground, 5 times per second during ascent and once per second during descent and landing.

The high-g accelerometer is reported separately from the data for the 9-axis IMU (accel/gyro/mag). The 9-axis IMU is mounted so that the X axis is "across" the board (along the short axis), the Y axis is "along" the board (along the long axis, with the high-g accelerometer) and the Z axis is "through" the board (perpendicular to the board). Rotation measurements are around the respective axis, so Y rotation measures the spin rate of the rocket while X and Z rotation measure the tilt rate.

The overall tilt angle of the rocket is computed by first measuring the orientation of the rocket on the pad using the 3 axis accelerometer, and then integrating the overall tilt rate from the 3 axis gyroscope to compute the total orientation change of the airframe since liftoff.

**Table 5. TeleMega IMU Sensor Packet Contents**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>uint8_t</td>
<td>orient</td>
<td>Angle from vertical in degrees</td>
</tr>
<tr>
<td>6</td>
<td>int16_t</td>
<td>accel</td>
<td>High G accelerometer</td>
</tr>
<tr>
<td>8</td>
<td>int32_t</td>
<td>pres</td>
<td>pressure (Pa * 10)</td>
</tr>
<tr>
<td>12</td>
<td>int16_t</td>
<td>temp</td>
<td>temperature (°C * 100)</td>
</tr>
<tr>
<td>14</td>
<td>int16_t</td>
<td>accel_x</td>
<td>X axis acceleration (across)</td>
</tr>
<tr>
<td>16</td>
<td>int16_t</td>
<td>accel_y</td>
<td>Y axis acceleration (along)</td>
</tr>
<tr>
<td>Offset</td>
<td>Data Type</td>
<td>Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>--------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>18</td>
<td>int16_t</td>
<td>accel_z</td>
<td>Z axis acceleration (through)</td>
</tr>
<tr>
<td>20</td>
<td>int16_t</td>
<td>gyro_x</td>
<td>X axis rotation (across)</td>
</tr>
<tr>
<td>22</td>
<td>int16_t</td>
<td>gyro_y</td>
<td>Y axis rotation (along)</td>
</tr>
<tr>
<td>24</td>
<td>int16_t</td>
<td>gyro_z</td>
<td>Z axis rotation (through)</td>
</tr>
<tr>
<td>26</td>
<td>int16_t</td>
<td>mag_x</td>
<td>X field strength (across)</td>
</tr>
<tr>
<td>28</td>
<td>int16_t</td>
<td>mag_y</td>
<td>Y field strength (along)</td>
</tr>
<tr>
<td>30</td>
<td>int16_t</td>
<td>mag_z</td>
<td>Z field strength (through)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>uint8_t</td>
<td>state</td>
<td>Flight state</td>
</tr>
<tr>
<td>6</td>
<td>int16_t</td>
<td>v_batt</td>
<td>battery voltage</td>
</tr>
<tr>
<td>8</td>
<td>int16_t</td>
<td>v_pyro</td>
<td>pyro battery voltage</td>
</tr>
<tr>
<td>10</td>
<td>int8_t[6]</td>
<td>sense</td>
<td>pyro continuity sense</td>
</tr>
<tr>
<td>16</td>
<td>int32_t</td>
<td>ground_pres</td>
<td>Average barometer reading on ground</td>
</tr>
<tr>
<td>20</td>
<td>int16_t</td>
<td>ground_accel</td>
<td>Average accelerometer reading on ground</td>
</tr>
<tr>
<td>22</td>
<td>int16_t</td>
<td>accel_plus_g</td>
<td>Accel calibration at +1g</td>
</tr>
<tr>
<td>24</td>
<td>int16_t</td>
<td>accel_minus_g</td>
<td>Accel calibration at -1g</td>
</tr>
<tr>
<td>26</td>
<td>int16_t</td>
<td>acceleration</td>
<td>m/s² * 16</td>
</tr>
<tr>
<td>28</td>
<td>int16_t</td>
<td>speed</td>
<td>m/s * 16</td>
</tr>
<tr>
<td>30</td>
<td>int16_t</td>
<td>height</td>
<td>m</td>
</tr>
</tbody>
</table>

Table 6. TeleMega Kalman and Voltage Data Packet Contents

2.4. TeleMetrum v2 and newer Sensor Data

Table 7. TeleMetrum v2 Packet Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0A</td>
<td>TeleMetrum v2 Sensor Data</td>
</tr>
<tr>
<td>0x0B</td>
<td>TeleMetrum v2 Calibration Data</td>
</tr>
</tbody>
</table>

TeleMetrum v2 and newer have higher resolution barometric data than TeleMetrum v1, and so the constant calibration data is split out into a separate packet.

TeleMetrum v2 and newer 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.

TeleMetrum v2 and newer Calibration Data packets are always transmitted once per second.

Table 8. TeleMetrum v2 and newer Sensor Packet Contents
Table 9. TeleMetrum v2 and newer Calibration Data Packet Contents

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>pad[3]</td>
<td>pad bytes</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>int32_t</td>
<td>ground_pres</td>
<td>Average barometer reading on ground</td>
</tr>
<tr>
<td>12</td>
<td>int16_t</td>
<td>ground_accel</td>
<td>Average accelerometer reading on ground</td>
</tr>
<tr>
<td>14</td>
<td>int16_t</td>
<td>accel_plus_g</td>
<td>Accel calibration at +1g</td>
</tr>
<tr>
<td>16</td>
<td>int16_t</td>
<td>accel_minus_g</td>
<td>Accel calibration at -1g</td>
</tr>
<tr>
<td>18</td>
<td>pad[14]</td>
<td>pad bytes</td>
<td></td>
</tr>
</tbody>
</table>

2.5. TeleMini v3.0 Sensor Data

Table 10. Sensor Packet Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x11</td>
<td>TeleMini v3.0 Sensor Data</td>
</tr>
</tbody>
</table>

TeleMini v3.0 uses this packet format for sensor data.

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 11. Sensor Packet Contents

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>uint8_t</td>
<td>state</td>
<td>Flight state</td>
</tr>
<tr>
<td>6</td>
<td>int16_t</td>
<td>v_batt</td>
<td>battery voltage</td>
</tr>
<tr>
<td>8</td>
<td>int16_t</td>
<td>sense_a</td>
<td>apogee continuity sense</td>
</tr>
<tr>
<td>10</td>
<td>int16_t</td>
<td>sense_m</td>
<td>main continuity sense</td>
</tr>
<tr>
<td>12</td>
<td>int32_t</td>
<td>pres</td>
<td>pressure sensor (Pa * 10)</td>
</tr>
</tbody>
</table>
### 2.6. Configuration Data

**Table 12. Configuration Packet Type**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x04</td>
<td>Configuration Data</td>
</tr>
</tbody>
</table>

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 13. Configuration Packet Contents**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>uint8_t</td>
<td>type</td>
<td>Device type</td>
</tr>
<tr>
<td>6</td>
<td>uint16_t</td>
<td>flight</td>
<td>Flight number</td>
</tr>
<tr>
<td>8</td>
<td>uint8_t</td>
<td>config_major</td>
<td>Config major version</td>
</tr>
<tr>
<td>9</td>
<td>uint8_t</td>
<td>config_minor</td>
<td>Config minor version</td>
</tr>
<tr>
<td>10</td>
<td>uint16_t</td>
<td>apogee_delay</td>
<td>Apogee deploy delay in seconds</td>
</tr>
<tr>
<td>12</td>
<td>uint16_t</td>
<td>main_deploy</td>
<td>Main deploy alt in meters</td>
</tr>
<tr>
<td>14</td>
<td>uint16_t</td>
<td>flight_log_max</td>
<td>Maximum flight log size (kB)</td>
</tr>
<tr>
<td>16</td>
<td>char</td>
<td>callsign[8]</td>
<td>Radio operator identifier</td>
</tr>
<tr>
<td>24</td>
<td>char</td>
<td>version[8]</td>
<td>Software version identifier</td>
</tr>
</tbody>
</table>

### 2.7. GPS Location

**Table 14. GPS Packet Type**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x05</td>
<td>GPS Location</td>
</tr>
</tbody>
</table>

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

GPS Location packets are transmitted once per second during all phases of the flight.
Table 15. GPS Location Packet Contents

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>uint8_t</td>
<td>flags</td>
<td>See GPS Flags table below</td>
</tr>
<tr>
<td>6</td>
<td>int16_t</td>
<td>altitude</td>
<td>m</td>
</tr>
<tr>
<td>8</td>
<td>int32_t</td>
<td>latitude</td>
<td>degrees * 107</td>
</tr>
<tr>
<td>12</td>
<td>int32_t</td>
<td>longitude</td>
<td>degrees * 107</td>
</tr>
<tr>
<td>16</td>
<td>uint8_t</td>
<td>year</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>uint8_t</td>
<td>month</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>uint8_t</td>
<td>day</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>uint8_t</td>
<td>hour</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>uint8_t</td>
<td>minute</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>uint8_t</td>
<td>second</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>uint8_t</td>
<td>pdop</td>
<td>* 5</td>
</tr>
<tr>
<td>23</td>
<td>uint8_t</td>
<td>hdop</td>
<td>* 5</td>
</tr>
<tr>
<td>24</td>
<td>uint8_t</td>
<td>vdop</td>
<td>* 5</td>
</tr>
<tr>
<td>25</td>
<td>uint8_t</td>
<td>mode</td>
<td>See GPS Mode table below</td>
</tr>
<tr>
<td>26</td>
<td>uint16_t</td>
<td>ground_speed</td>
<td>cm/s</td>
</tr>
<tr>
<td>28</td>
<td>int16_t</td>
<td>climb_rate</td>
<td>cm/s</td>
</tr>
<tr>
<td>30</td>
<td>uint8_t</td>
<td>course</td>
<td>/ 2</td>
</tr>
<tr>
<td>31</td>
<td>uint8_t</td>
<td>unused[1]</td>
<td></td>
</tr>
</tbody>
</table>

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 16. GPS Flags

<table>
<thead>
<tr>
<th>Bits</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-3</td>
<td>nsats</td>
<td>Number of satellites in solution</td>
</tr>
<tr>
<td>4</td>
<td>valid</td>
<td>GPS solution is valid</td>
</tr>
<tr>
<td>5</td>
<td>running</td>
<td>GPS receiver is operational</td>
</tr>
<tr>
<td>6</td>
<td>date_valid</td>
<td>Reported date is valid</td>
</tr>
<tr>
<td>7</td>
<td>course_valid</td>
<td>ground speed, course and climb rates are valid</td>
</tr>
</tbody>
</table>

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 17. GPS Mode
Mode | Name                  | Description                                                                 |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Not Valid</td>
<td>All data are invalid</td>
</tr>
<tr>
<td>A</td>
<td>Autonomous mode</td>
<td>Data are derived from satellite data</td>
</tr>
<tr>
<td>D</td>
<td>Differential Mode</td>
<td>Data are augmented with differential data from a known ground station. The SkyTraq unit in TeleMetrurm does not support this mode</td>
</tr>
<tr>
<td>E</td>
<td>Estimated</td>
<td>Data are estimated using dead reckoning from the last known data</td>
</tr>
<tr>
<td>M</td>
<td>Manual</td>
<td>Data were entered manually</td>
</tr>
<tr>
<td>S</td>
<td>Simulated</td>
<td>GPS receiver testing mode</td>
</tr>
</tbody>
</table>

2.8. GPS Satellite Data

Table 18. GPS Satellite Data Packet Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x06</td>
<td>GPS Satellite Data</td>
</tr>
</tbody>
</table>

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 19. GPS Satellite Data Contents

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>uint8_t</td>
<td>channels</td>
<td>Number of reported satellite information</td>
</tr>
<tr>
<td>6</td>
<td>sat_info_t</td>
<td>sats[12]</td>
<td>See Per-Satellite data table below</td>
</tr>
<tr>
<td>30</td>
<td>uint8_t</td>
<td>unused[2]</td>
<td></td>
</tr>
</tbody>
</table>

Table 20. GPS Per-Satellite data (sat_info_t)

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>uint8_t</td>
<td>svid</td>
<td>Space Vehicle Identifier</td>
</tr>
<tr>
<td>1</td>
<td>uint8_t</td>
<td>c_n_1</td>
<td>C/N1 signal quality indicator</td>
</tr>
</tbody>
</table>

2.9. Companion Data

Table 21. Companion Data Packet Type

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x07</td>
<td>Companion Data</td>
</tr>
</tbody>
</table>

When a companion board is attached to TeleMega or TeleMetrurum, it can provide telemetry data to be included in the downlink. The companion board can provide up to 12 16-bit data values.
The companion board itself specifies the transmission rate. On the ground and during descent, that rate is limited to one packet per second. During ascent, that rate is limited to 10 packets per second.

<table>
<thead>
<tr>
<th>Offset</th>
<th>Data Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>uint8_t</td>
<td>board_id</td>
<td>Type of companion board attached</td>
</tr>
<tr>
<td>6</td>
<td>uint8_t</td>
<td>update_period</td>
<td>How often telemetry is sent, in 1/100ths of a second</td>
</tr>
<tr>
<td>7</td>
<td>uint8_t</td>
<td>channels</td>
<td>Number of data channels supplied</td>
</tr>
<tr>
<td>8</td>
<td>uint16_t[12]</td>
<td>companion_data</td>
<td>Up to 12 channels of 16-bit companion data</td>
</tr>
</tbody>
</table>

### 3. Data Transmission

Altus Metrum devices use Texas Instruments sub-GHz digital radio products. Ground stations use parts with HW FEC while some flight computers perform FEC in software. TeleGPS is transmit-only.

<table>
<thead>
<tr>
<th>Part Number</th>
<th>Description</th>
<th>Used in</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC1111</td>
<td>10mW transceiver with integrated SoC</td>
<td>TeleDongle v0.2, TeleBT v1.0, TeleMetrum v1.x, TeleMini v1</td>
</tr>
<tr>
<td>CC1120</td>
<td>35mW transceiver with SW FEC</td>
<td>TeleMetrum v2, TeleMega v1</td>
</tr>
<tr>
<td>CC1200</td>
<td>35mW transceiver with HW FEC</td>
<td>TeleMetrum v3, TeleMega v2, TeleDongle v3.0, TeleMini v3, TeleBT v3.0, TeleGPS v2</td>
</tr>
<tr>
<td>CC115L</td>
<td>14mW transmitter with SW FEC</td>
<td>TeleGPS v1</td>
</tr>
</tbody>
</table>

### 3.1. Modulation Scheme

Texas Instruments provides a tool for computing modulation parameters given a desired modulation format and basic bit rate.

While we might like to use something with better low-signal performance like BPSK, the radios we use don't support that, but do support Gaussian frequency shift keying (GFSK). Regular frequency shift keying (FSK) encodes the signal by switching the carrier between two frequencies. The Gaussian version is essentially the same, but the shift between frequencies gently follows a gaussian curve, rather than switching immediately. This tames the bandwidth of the signal without affecting the ability to transmit data.

For AltOS, there are three available bit rates, 38.4kBaud, 9.6kBaud and 2.4kBaud resulting in the following signal parameters:
<table>
<thead>
<tr>
<th>Rate</th>
<th>Deviation</th>
<th>Receiver Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>38.4kBaud</td>
<td>20.5kHz</td>
<td>100kHz</td>
</tr>
<tr>
<td>9.6kBaud</td>
<td>5.125kHz</td>
<td>25kHz</td>
</tr>
<tr>
<td>2.4kBaud</td>
<td>1.5kHz</td>
<td>5kHz</td>
</tr>
</tbody>
</table>

### 3.2. Error Correction

The cc1111 and cc1200 provide forward error correction in hardware; on the cc1120 and cc115l that's done in software. AltOS uses this to improve reception of weak signals. As it's a rate 1/2 encoding, each bit of data takes two bits when transmitted, so the effective data rate is half of the raw transmitted bit rate.

Table 25. Error Correction

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Correction</td>
<td>Convolutional coding</td>
<td>1/2 rate, constraint length m=4</td>
</tr>
<tr>
<td>Interleaving</td>
<td>4 x 4</td>
<td>Reduce effect of noise burst</td>
</tr>
<tr>
<td>Data Whitening</td>
<td>XOR with 9-bit PNR</td>
<td>Rotate right with bit 8 = bit 0 xor bit 5, initial value 11111111</td>
</tr>
</tbody>
</table>

### 4. TeleDongle serial 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**

```
224f01080b05765e00701f1a1bb8b8d7b60b070605140c000600000000000000003fa988
```

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 26. TeleDongle serial Packet Format

<table>
<thead>
<tr>
<th>Offset</th>
<th>Name</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>length</td>
<td>22</td>
<td>Total length of data bytes in the line. Note that this includes the added RSSI and status bytes</td>
</tr>
<tr>
<td>1 ·· length-3</td>
<td>packet</td>
<td>4f ·· 00</td>
<td>Bytes of actual packet data</td>
</tr>
<tr>
<td>length-2</td>
<td>rssi</td>
<td>3f</td>
<td>Received signal strength. dBM = rssi / 2 - 74</td>
</tr>
</tbody>
</table>
5. 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.