EasyMini Owner's Manual

Keith Packard, Bdale Garbee

Table of Contents

License

Acknowledgments

Thanks to Bob Finch, W9YA, NAR 12965, TRA 12350 for writing "The Mere-Mortals Quick Start/Usage Guide to the Altus Metrum Starter Kit" which formed the basis of the original Getting Started chapter in this manual. Bob was one of our first customers for a production TeleMetrum, and his continued enthusiasm and contributions are immensely gratifying and highly appreciated!

And thanks to Anthony (AJ) Towns for major contributions including the AltosUI graphing and site map code and associated documentation. Free software means that our customers and friends can become our collaborators, and we certainly appreciate this level of contribution!

Have fun using these products, and we hope to meet all of you out on the rocket flight line somewhere.

> Bdale Garbee, KB0G
> NAR #87103, TRA #12201

> Keith Packard, KD7SQG
> NAR #88757, TRA #12200

Chapter 1. Introduction and Overview

Welcome to the Altus Metrum community! Our circuits and software reflect our passion for both hobby rocketry and Free Software. We hope their capabilities and performance will delight you in every way, but by releasing all of our hardware and software designs under open licenses, we also hope to empower you to take as active a role in our collective future as you wish!

Our goal is to include in this document all of the information required to successfully configure and use Altus Metrum products. But documentation is a lot like software in that it can contain "bugs", and can probably always be improved! If you have questions that aren't answered in this manual, or just need a little help figuring things out, we strongly suggest joining the Altus Metrum user email list, which you can do by visiting https://lists.gag.com/mailman/listinfo/altusmetrum. There's a lot of useful information in the mailing list archives!

The first device created for our community was TeleMetrum, a dual deploy altimeter with fully integrated GPS and radio telemetry as standard features, and a "companion interface" that will support optional capabilities in the future. The later versions of TeleMetrum, v2 and newer, have all of the same features but with improved sensors and radio to offer increased performance.

Our second device was TeleMini, a dual deploy altimeter with radio telemetry and radio direction finding. The first version of this device was only 13mm by 38mm (½ inch by 1½ inches) and could fit easily in an 18mm air-frame. The latest version, v3.0, includes a beeper, higher power radio, extended on-board flight logging and an improved barometric sensor.

TeleMega is our most sophisticated device, including six pyro channels (four of which are fully programmable), integrated GPS, integrated gyroscopes for staging/air-start inhibit and high performance telemetry.

EasyMini is a dual-deploy altimeter with logging and built-in USB data download.

EasyMega is essentially a TeleMega board with the GPS receiver and telemetry transmitter removed. It offers the same 6 pyro channels and integrated gyroscopes for staging/air-start inhibit.

TeleDongle v0.2 was our first ground station, providing a USB to RF interfaces for communicating with the altimeters. Combined with your choice of antenna and notebook computer, TeleDongle and our associated user interface software form a complete ground station capable of logging and displaying in-flight telemetry, aiding rocket recovery, then processing and archiving flight data for analysis and review. The latest version, TeleDongle v3, has all new electronics with a higher performance radio for improved range.

For a slightly more portable ground station experience that also provides direct rocket recovery support, TeleBT offers flight monitoring and data logging using a Bluetooth™ connection between the receiver and an Android device that has the AltosDroid application installed from the Google Play store.

More products will be added to the Altus Metrum family over time, and we currently envision that this will be a single, comprehensive manual for the entire product family.

Chapter 2. Getting Started

The first thing to do after you open the box is to hook up a battery and charge it if necessary.

2.1. Batteries

The Lithium Polymer EasyMini battery can be charged by disconnecting it from the board and plugging it into a standalone battery charger such as LipoCharger, and connecting that via a USB cable to a laptop or other USB power source.

You can also choose to use another battery with EasyMini, anything supplying between 4 and 12 volts should work fine (like a standard 9V battery), but if you are planning to fire pyro charges, ground testing is required to verify that the battery supplies enough current to fire your chosen e-matches.

2.2. Linux/Mac/Windows Ground Station Software

Next you should obtain and install the AltOS software. The AltOS distribution includes the AltosUI ground station program, current firmware images for all of the hardware, and a number of standalone utilities that are rarely needed. Pre-built binary packages are available for Linux, Microsoft Windows, Mac OSX. Full source code and build instructions are also available. The latest version may always be downloaded from http://altusmetrum.org/AltOS

Chapter 3. Using Altus Metrum Hardware

Here are general instructions for hooking up an Altus Metrum flight computer. Instructions specific to each model will be found in the section devoted to that model below.

3.1. Wiring and Electrical Interference

To prevent electrical interference from affecting the operation of the flight computer, it's important to always twist pairs of wires connected to the board. Twist the switch leads, the pyro leads and the battery leads. This reduces interference through a mechanism called common mode rejection.

3.2. Hooking Up Lithium Polymer Batteries

All Altus Metrum flight computers have a two pin JST PH series connector to connect up a single-cell Lithium Polymer cell (3.7V nominal). You can purchase matching batteries from the Altus Metrum store, or other vendors, or you can make your own. Pin 1 of the connector is positive, pin 2 is negative. Spark Fun sells a cable with the connector attached, which they call a JST Jumper 2 Wire Assembly

WARNING	Many RC vendors also sell lithium polymer batteries with this same connector. All that we have found use the opposite polarity, and if you use them that way, you will damage or destroy the flight computer.

3.3. Hooking Up Pyro Charges

Altus Metrum flight computers always have two screws for each pyro charge. This means you shouldn't need to put two wires into a screw terminal or connect leads from pyro charges together externally.

On the flight computer, one lead from each charge is hooked to the positive battery terminal through the power switch. The other lead is connected through the pyro circuit, which is connected to the negative battery terminal when the pyro circuit is fired.

3.4. Hooking Up a Power Switch

Altus Metrum flight computers need an external power switch to turn them on. This disconnects both the computer and the pyro charges from the battery, preventing the charges from firing when in the Off position. The switch is in-line with the positive battery terminal.

3.5. Understanding Beeps

Altus Metrum flight computers include a beeper to provide information about the state of the system.

Here's a short summary of all of the modes and the beeping that accompanies each mode. In

---

the description of the beeping pattern, "dit" means a short beep while "dah" means a long beep (three times as long). "Brap" means a long dissonant tone.

Table 1. AltOS Modes

Mode Name	Abbreviation	Beeps	Description
Startup	S	battery voltage in decivolts	Calibrating sensors, detecting orientation.
Idle	I	dit dit	Ready to accept commands over USB
Pad	P	dit dah dah dit	Waiting for launch. Not listening for commands.
Boost	B	dah dit dit dit	Accelerating upwards.
Fast	F	dit dit dah dit	Decelerating, but moving faster than 200m/s.
Coast	C	dah dit dah dit	Decelerating, moving slower than 200m/s
Drogue	D	dah dit dit	Descending after apogee. Above main height.
Main	M	dah dah	Descending. Below main height.
Landed	L	dit dah dit dit	Stable altitude for at least ten seconds.
Sensor error	X	dah dit dit dah	Error detected during sensor calibration.

Here's a summary of all of the Pad and Idle mode indications. In Idle mode, you'll hear one of these just once after the two short dits indicating idle mode. In Pad mode, after the dit dah dah dit indicating Pad mode, you'll hear these once every five seconds.

Table 2. Pad/Idle Indications

Name	Beeps	Description
Neither	brap	No continuity detected on either apogee or main igniters.
Apogee	dit	Continuity detected only on apogee igniter.
Main	dit dit	Continuity detected only on main igniter.
Both	dit dit dit	Continuity detected on both igniters.
Storage Full	warble	On-board data logging storage is full. This will not prevent the flight computer from safely controlling the flight or transmitting telemetry signals, but no record of the flight will be stored in on-board flash.

3.6. Turning On the Power

Connect a battery and power switch and turn the switch to "on". The flight computer will signal power on by reporting the battery voltage and then perform an internal self test and sensor calibration.

Once the self test and calibration are complete, there are two modes that an Altus Metrum flight computer can operate in:

Flight/Pad

> The flight computer is waiting to detect launch and then fly the rocket. In this mode, the USB link is disabled. The only way to get out of this mode is to power the flight computer down. See below for how to get the flight computer to come up in Flight/Pad mode at power on.

Idle

> The flight computer is ready to communicate over USB You can configure the flight computer, download data or display the current state. See below for how to get the flight computer to come up in Idle mode at power on.

For EasyMini, if the USB cable is connected to a computer, it will enter Idle mode. Otherwise, it will enter Flight/Pad mode.

You can see in Understanding Beeps how to tell which mode the flight computer is in.

3.7. Using an External Active Switch Circuit

You can use an active switch circuit, such as the Featherweight Magnetic Switch, with any Altus Metrum flight computer. These require three connections, one to the battery, one to the positive power input on the flight computer and one to ground. Find instructions on how to hook these up for each flight computer below. Then follow the instructions that come with your active switch to connect it up.

3.8. Using a Separate Pyro Battery

As mentioned above in Hooking Up Pyro Charges, one lead for each of the pyro charges is connected through the power switch directly to the positive battery terminal. The other lead is connected to the pyro circuit, which connects it to the negative battery terminal when the pyro circuit is fired. The pyro circuit on all of the flight computers is designed to handle up to 16V.

To use a separate pyro battery, connect the negative pyro battery terminal to the flight computer ground terminal, the the switched positive battery terminal to the igniter and the other igniter lead to the negative pyro terminal on the flight computer. When the pyro channel fires, it will complete the circuit between the negative pyro terminal and the ground terminal, firing the igniter. Specific instructions on how to hook this up for each flight computer will be found in the section below for that flight computer.

Note that you must include a switch in the positive lead of the pyro battery for safety, as the on-

board power switch circuit on our product only controls current flow from the the primary or system battery!

3.9. Using a Different Kind of Battery

EasyMini is designed to use either a lithium polymer battery or any other battery producing between 4 and 12 volts, such as a rectangular 9V battery.

3.10. Using Packet Link Mode

All AltusMetrum flight computers that have a radio can communicate with the ground station software for configuration and other operations using the Packet Link mode. This uses radio communication instead of a USB cable. To set this up, the ground station software must be configured to the correct data rate, frequency and callsign.

You can monitor Packet Link mode from TeleBT or TeleDongle by watching the LEDs. Each time the device transmits, the red LED will flash. When the link is busy, or when the link is not working, the device will transmit 10 times per second, so the LED will flash rapidly. When the link is working and there is no data to send, the link will flash once per second, and the LED will flash more slowly.

Chapter 4. EasyMini
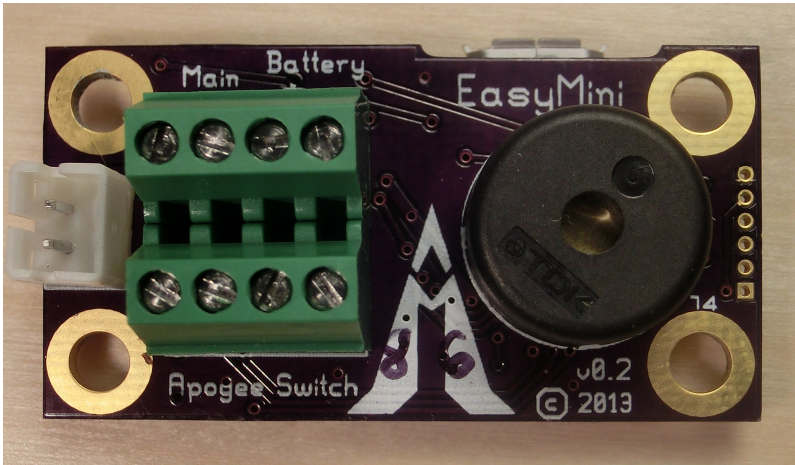
Figure 1. EasyMini Board

EasyMini is built on a 0.8 inch by 1½ inch circuit board. It's designed to fit in a 24mm coupler tube.

You usually don't need to configure EasyMini at all; it's set up to do dual-deployment with an event at apogee to separate the airframe and deploy a drogue and another event at 250m (820ft) to deploy the main. Install EasyMini in your airframe, hook up a battery, igniters and a power switch and you're ready to fly.

4.1. EasyMini Screw Terminals

EasyMini has two sets of four screw terminals near one end of the board. Using the picture above, the top four have connections for the main pyro circuit and an external battery and the bottom four have connections for the apogee pyro circuit and the power switch. Counting from the left, the connections are as follows:

Table 3. EasyMini Screw Terminals

Terminal #	Terminal Name	Description
Top 1	Main -	Main pyro channel connection to pyro circuit
Top 2	Main +	Main pyro channel common connection to battery +
Top 3	Battery +	Positive external battery terminal
Top 4	Battery -	Negative external battery terminal
Bottom 1	Apogee -	Apogee pyro channel connection to pyro circuit
Bottom 2	Apogee +	Apogee pyro channel common connection to battery +
Bottom 3	Switch Output	Switch connection to flight computer
Bottom 4	Switch Input	Switch connection to positive battery terminal

4.2. Connecting A Battery To EasyMini

There are two possible battery connections on EasyMini. You can use either method; both feed through the power switch terminals.

One battery connection is the standard Altus Metrum white JST plug. This mates with single-cell Lithium Polymer batteries sold by Altus Metrum.

The other is a pair of screw terminals marked 'Battery +' and 'Battery -'. Connect a battery from 4 to 12 volts to these terminals, being careful to match polarity.

4.3. Charging Lithium Batteries

Because EasyMini allows for batteries other than the standard Altus Metrum Lithium Polymer cells, it cannot incorporate a battery charger circuit. Therefore, when using a Litium Polymer cell, you'll need an external charger. These are available from Altus Metrum, or from Spark Fun.

4.4. Using a Separate Pyro Battery with EasyMini

As described above, using an external pyro battery involves connecting the negative battery terminal to the flight computer ground, connecting the positive battery terminal to one of the igniter leads and connecting the other igniter lead to the per-channel pyro circuit connection.

To connect the negative pyro battery terminal to EasyMini ground, connect it to the negative external battery connection, top terminal 4.

Connecting the switched positive battery terminal to the pyro charges must be done separate from EasyMini, by soldering them together or using some other connector. Note that for safety, you must put a switch between the pyro battery and the rest of the circuit!

The other lead from each pyro charge is then inserted into the appropriate per-pyro channel screw terminal (top terminal 1 for the Main charge, bottom terminal 1 for the Apogee charge).

4.5. Using an Active Switch with EasyMini

As explained above, an external active switch requires three connections, one to the positive battery terminal, one to the flight computer positive input and one to ground. Use the negative external battery connection, top terminal 4 for ground.

The positive battery terminal is available on bottom terminal 4, the positive flight computer input is on the bottom terminal 3.

Chapter 5. Installation

A typical installation involves attaching only a suitable battery, a single pole switch for power on/off, and two pairs of wires connecting e-matches for the apogee and main ejection charges. All Altus Metrum products are designed for use with single-cell batteries with 3.7 volts nominal. EasyMini may also be used with other batteries as long as they supply between 4 and 12 volts.

The battery connectors are a standard 2-pin JST connector; you can purchase suitable batteries from the any vendor selling Altus Metrum products. These batteries are single-cell Lithium Polymer batteries that nominally provide 3.7 volts. Other vendors sell similar batteries for RC aircraft using mating connectors, however the polarity for those is generally reversed from the batteries used by Altus Metrum products. In particular, the Tenergy batteries supplied for use in Featherweight flight computers are not compatible with Altus Metrum flight computers or battery chargers.

WARNING	Check polarity and voltage before connecting any battery not purchased from Altus Metrum.
WARNING	Spark Fun sells batteries that have a matching connector with the correct polarity. However, these batteries include an integrated current limiting circuit. That circuit will cause the battery to shut down when firing the igniter circuit. Do not use these batteries unless you remove the current limiting circuit.

By default, we use the unregulated output of the battery directly to fire ejection charges. This works marvelously with standard low-current e-matches like the J-Tek from MJG Technologies, and with Quest Q2G2 igniters. However, if you want or need to use a separate pyro battery, check out Using a Separate Pyro Battery for instructions on how to wire that up. The altimeters are designed to work with an external pyro battery of no more than 15 volts.

Ejection charges are wired directly to the screw terminal block at the aft end of the altimeter. You'll need a very small straight blade screwdriver for these screws, such as you might find in a jeweler's screwdriver set. The screw terminal block is also used for the power switch leads.

Chapter 6. Using Altus Metrum Products

6.1. In the Rocket

In the rocket itself, you just need a flight computer and a single-cell, 3.7 volt nominal Li-Po rechargeable battery. A 110mAh battery weighs less than a triple A battery and is a good choice for use with EasyMini.

6.2. On the Ground

The GUI tool, AltosUI, is written in Java and runs across Linux, Mac OS and Windows. There's also a suite of C tools for Linux which can perform most of the same tasks.

After the flight, you can use a USB cable to plug into the flight computer board directly. A USB cable is also how you charge the Li-Po battery, so you'll want one of those anyway. The same cable used by lots of digital cameras and other modern electronic stuff will work fine.

6.3. Data Analysis

Our software makes it easy to log the data from each flight, both the telemetry received during the flight itself, and the more complete data log recorded in the flash memory on the altimeter board. Once this data is on your computer, our post-flight tools make it easy to quickly get to the numbers everyone wants, like apogee altitude, max acceleration, and max velocity. You can also generate and view a standard set of plots showing the altitude, acceleration, and velocity of the rocket during flight. And you can even export a flight log in a format usable with Google Maps and Google Earth for visualizing the flight path in two or three dimensions!

Our ultimate goal is to emit a set of files for each flight that can be published as a web page per flight, or just viewed on your local disk with a web browser.

6.4. Future Plans

Because all of our work is open, both the hardware designs and the software, if you have some great idea for an addition to the current Altus Metrum family, feel free to dive in and help! Or let us know what you'd like to see that we aren't already working on, and maybe we'll get excited about it too...

Watch our web site for more news and information as our family of products evolves!
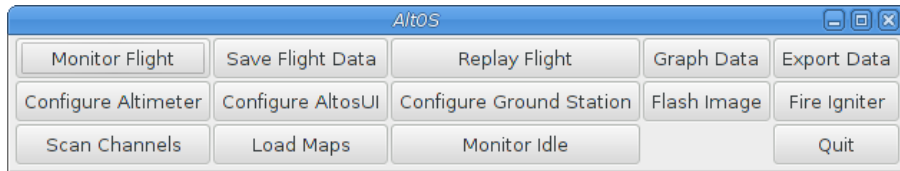
Chapter 7. AltosUI

Figure 2. AltosUI Main Window

The AltosUI program provides a graphical user interface for interacting with the Altus Metrum product family. AltosUI can monitor telemetry data, configure devices and many other tasks. The primary interface window provides a selection of buttons, one for each major activity in the system. This chapter is split into sections, each of which documents one of the tasks provided from the top-level toolbar.

7.1. Save Flight Data

The altimeter records flight data to its internal flash memory. The 'Save Flight Data' button allows you to read the flash memory and write it to disk.

Clicking on the 'Save Flight Data' button brings up a list of connected flight computers and TeleDongle devices. If you select a flight computer, the flight data will be downloaded from that device directly.

After the device has been selected, a dialog showing the flight data saved in the device will be shown allowing you to select which flights to download and which to delete. With version 0.9 or newer firmware, you must erase flights in order for the space they consume to be reused by another flight. This prevents accidentally losing flight data if you neglect to download data before flying again. Note that if there is no more space available in the device, then no data will be recorded during the next flight.

The file name for each flight log is computed automatically from the recorded flight date, altimeter serial number and flight number information.

7.2. Replay Flight

Select this button and you are prompted to select a flight record file, either a .telem file recording telemetry data or a .eeprom file containing flight data saved from the altimeter flash memory.

Once a flight record is selected, the flight monitor interface is displayed and the flight is re-enacted in real time.

7.3. Graph Data

Select this button and you are prompted to select a flight record file, either a .telem file recording telemetry data or a .eeprom file containing flight data saved from flash memory.

Note that telemetry files will generally produce poor graphs due to the lower sampling rate and missed telemetry packets. Use saved flight data in .eeprom files for graphing where possible.

Once a flight record is selected, a window with multiple tabs is opened.

7.3.1. Flight Graph
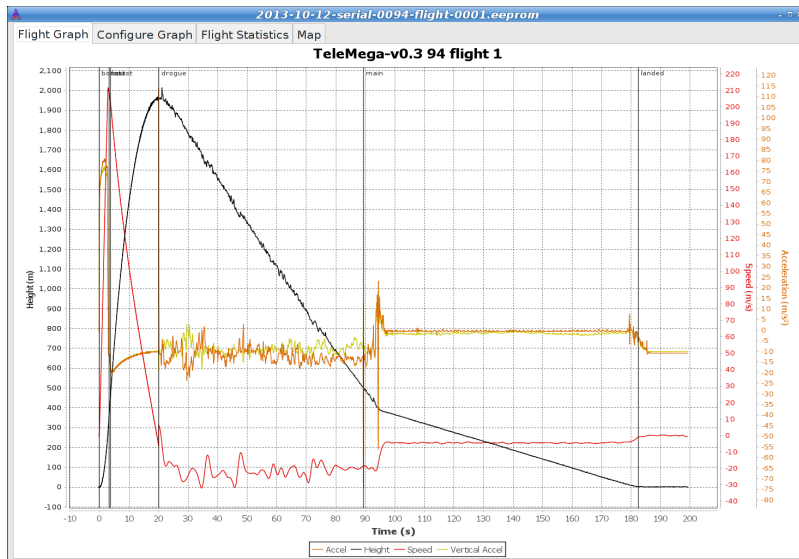
Figure 3. Flight Data Graph

By default, the graph contains acceleration (blue), velocity (green) and altitude (red).

The graph can be zoomed into a particular area by clicking and dragging down and to the right. Once zoomed, the graph can be reset by clicking and dragging up and to the left. Holding down control and clicking and dragging allows the graph to be panned. The right mouse button causes a pop-up menu to be displayed, giving you the option save or print the plot.

7.3.2. Configure Graph
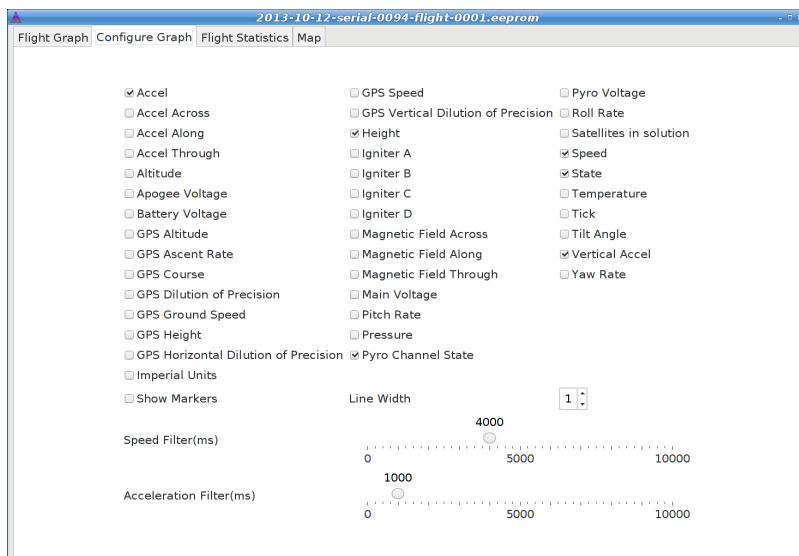
Figure 4. Flight Graph Configuration

This selects which graph elements to show, and, at the very bottom. It also lets you configure how the graph is drawn:

- Whether to use metric or imperial units
- Whether to show a marker at each data point. When displaying a small section of the graph, these can be useful to know what data values were recorded.
- How wide to draw the lines in the graph
- How to filter speed and acceleration data computed from barometric data. Flight computers with accelerometers never display computed acceleration data, and only use barometric data to compute speed during descent. Flight computers without accelerometers always compute both speed and acceleration from barometric data. A larger value smooths the data more.

7.3.3. Flight Statistics

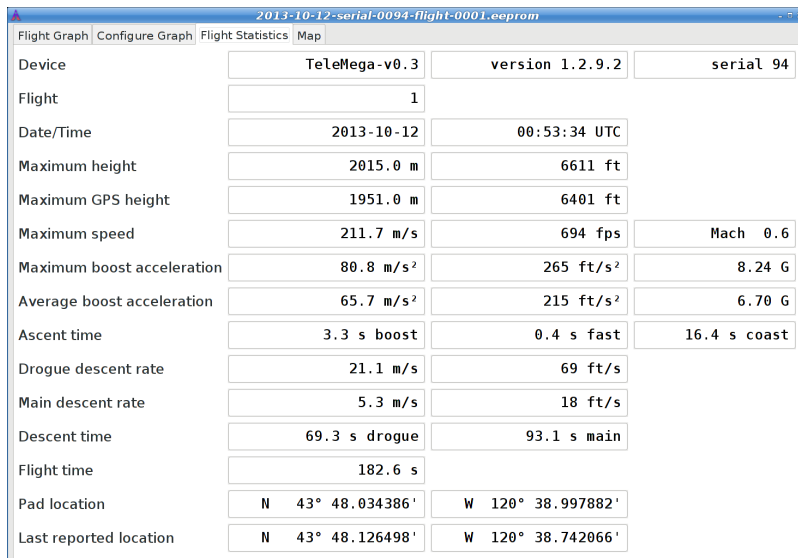|			
Device	TeleMega-v0.3	version 1.2.9.2	serial 94
Flight	1		
Date/Time	2013-10-12	00:53:34 UTC	
Maximum height	2015.0 m	6611 ft	
Maximum GPS height	1951.0 m	6401 ft	
Maximum speed	211.7 m/s	694 fps	Mach 0.6
Maximum boost acceleration	80.8 m/s²	265 ft/s²	8.24 G
Average boost acceleration	65.7 m/s²	215 ft/s²	6.70 G
Ascent time	3.3 s boost	0.4 s fast	16.4 s coast
Drogue descent rate	21.1 m/s	69 ft/s	
Main descent rate	5.3 m/s	18 ft/s	
Descent time	69.3 s drogue	93.1 s main	
Flight time	182.6 s		
Pad location	N 43° 48.034386'	W 120° 38.997882'	
Last reported location	N 43° 48.126498'	W 120° 38.742066'	

Figure 5. Flight Statistics

Shows overall data computed from the flight.

7.4. Export Data

This tool takes the raw data files and makes them available for external analysis. When you select this button, you are prompted to select a flight data file, which can be either a .eeprom or .telem. The .eeprom files contain higher resolution and more continuous data, while .telem files contain receiver signal strength information. Next, a second dialog appears which is used to select where to write the resulting file.
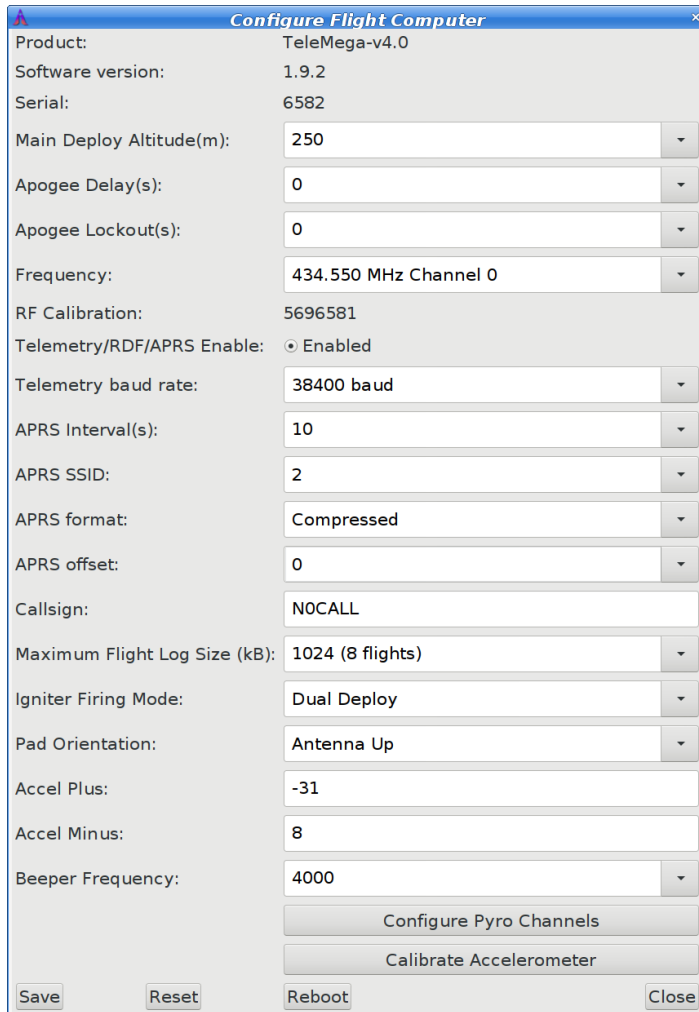
7.4.1. Comma Separated Value Format

This is a text file containing the data in a form suitable for import into a spreadsheet or other external data analysis tool. The first few lines of the file contain the version and configuration information from the altimeter, then there is a single header line which labels all of the fields. All

of these lines start with a '#' character which many tools can be configured to skip over.

The remaining lines of the file contain the data, with each field separated by a comma and at least one space. All of the sensor values are converted to standard units, with the barometric data reported in both pressure, altitude and height above pad units.

7.5. Configure Altimeter

Figure 6. Altimeter Configuration

Select this button and then select an altimeter.

The first few lines of the dialog provide information about the connected device, including the product name, software version and hardware serial number. Below that are the individual configuration entries.

At the bottom of the dialog, there are four buttons:

Save

> This writes any changes to the configuration parameter block in flash memory. If you don't press this button, any changes you make will be lost.

Reset

> This resets the dialog to the most recently saved values, erasing any changes you have made.

Reboot

> This reboots the device. Use this to switch from idle to pad mode by rebooting once the rocket is oriented for flight, or to confirm changes you think you saved are really saved.

Close

> This closes the dialog. Any unsaved changes will be lost.

The rest of the dialog contains the parameters to be configured.

7.5.1. Main Deploy Altitude

This sets the altitude (above the recorded pad altitude) at which the 'main' igniter will fire. The drop-down menu shows some common values, but you can edit the text directly and choose whatever you like. If the apogee charge fires below this altitude, then the main charge will fire two seconds after the apogee charge fires.

7.5.2. Apogee Delay

When flying redundant electronics, it's often important to ensure that multiple apogee charges don't fire at precisely the same time, as that can over pressurize the apogee deployment bay and cause a structural failure of the air-frame. The Apogee Delay parameter tells the flight computer to fire the apogee charge a certain number of seconds after apogee has been detected.

7.5.3. Apogee Lockout

Apogee lockout is the number of seconds after launch where the flight computer will not fire the apogee charge, even if the rocket appears to be at apogee. This is often called 'Mach Delay', as it is intended to prevent a flight computer from unintentionally firing apogee charges due to the pressure spike that occurrs across a mach transition. Altus Metrum flight computers include a Kalman filter which is not fooled by this sharp pressure increase, and so this setting should be left at the default value of zero to disable it.

WARNING | Firmware versions older than 1.8.6 have a bug which resets the time since launch to zero each time a motor starts burning. Update firmware to get the correct behavior.

7.5.4. Maximum Flight Log Size

This sets the space (in kilobytes) allocated for each flight log. The available space will be divided into chunks of this size. A smaller value will allow more flights to be stored, a larger value will record data from longer flights.

7.5.5. Ignitor Firing Mode

This configuration parameter allows the two standard ignitor channels (Apogee and Main) to be used in different configurations.

Dual Deploy

This is the usual mode of operation; the 'apogee' channel is fired at apogee and the 'main' channel at the height above ground specified by the 'Main Deploy Altitude' during descent.

Redundant Apogee

This fires both channels at apogee, the 'apogee' channel first followed after a two second delay by the 'main' channel.

Redundant Main

This fires both channels at the height above ground specified by the Main Deploy Altitude setting during descent. The 'apogee' channel is fired first, followed after a two second delay by the 'main' channel.

Separation & Apogee

This fires the 'main' channel when the first motor burns out and fires the 'apogee' charge at apogee.

7.5.6. Beeper Frequency

The beeper on all Altus Metrum flight computers works best at 4000Hz, however if you have more than one flight computer in a single airframe, having all of them sound at the same frequency can be confusing. This parameter lets you adjust the base beeper frequency value.

7.6. Configure AltosUI
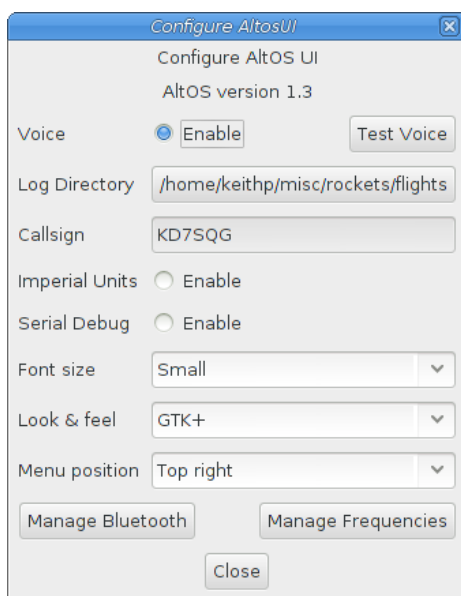
Figure 7. Configure AltosUI Dialog

This button presents a dialog so that you can configure the AltosUI global settings.

7.6.1. Log Directory

AltosUI logs all telemetry data and saves all flash data to this directory. This directory is also used as the staring point when selecting data files for display or export.

Click on the directory name to bring up a directory choosing dialog, select a new directory and click 'Select Directory' to change where AltosUI reads and writes data files.

7.6.2. Imperial Units

This switches between metric units (meters) and imperial units (feet and miles). This affects the display of values use during flight monitoring, configuration, data graphing and all of the voice announcements. It does not change the units used when exporting to CSV files, those are always produced in metric units.

7.6.3. Serial Debug

This causes all communication with a connected device to be dumped to the console from which AltosUI was started. If you've started it from an icon or menu entry, the output will simply be discarded. This mode can be useful to debug various serial communication issues.

7.6.4. Font size

Selects the set of fonts used in the flight monitor window. Choose between the small, medium and large sets.

7.6.5. Look & feel

Switches between the available Java user interface appearances. The default selection is supposed to match the native window system appearance for the target platform.

7.6.6. Menu position

Selects the initial position for the main AltosUI window that includes all of the command buttons.

7.7. Flash Image

This reprograms Altus Metrum devices with new firmware. EasyMini is programmed directly over USB (self programming). Please read the directions for flashing devices in Updating Device Firmware.

7.8. Fire Igniter
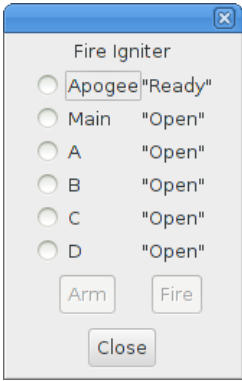
Figure 8. Fire Igniter Window

This activates the igniter circuits in the flight computer to help test recovery systems deployment.

Selecting the 'Fire Igniter' button brings up the usual device selection dialog. Pick the desired device. This brings up another window which shows the current continuity test status for all of the pyro channels.

Next, select the desired igniter to fire. This will enable the 'Arm' button.

Select the 'Arm' button. This enables the 'Fire' button. The word 'Arm' is replaced by a countdown timer indicating that you have 10 seconds to press the 'Fire' button or the system will deactivate, at which point you start over again at selecting the desired igniter.

Appendix A: System Operation

A.1. Firmware Modes

The AltOS firmware build for the altimeters has two fundamental modes, "idle" and "flight". Which of these modes the firmware operates in is determined at start up time. Since EasyMini doesn't have an accelerometer we can use to determine orientation, "idle" mode is selected if the board is connected via USB to a computer, otherwise the board enters "flight" mode.

At power on, the altimeter will beep out the battery voltage to the nearest tenth of a volt. Each digit is represented by a sequence of short "dit" beeps, with a pause between digits. A zero digit is represented with one long "dah" beep. Then there will be a short pause while the altimeter completes initialization and self test, and decides which mode to enter next.

In flight or "pad" mode, the altimeter engages the flight state machine, goes into transmit-only mode to send telemetry, and waits for launch to be detected. Flight mode is indicated by an "di-dah-dah-dit" ("P" for pad) on the beeper or lights, followed by beeps or flashes indicating the state of the pyrotechnic igniter continuity. One beep/flash indicates apogee continuity, two beeps/flashes indicate main continuity, three beeps/flashes indicate both apogee and main continuity, and one longer "brap" sound which is made by rapidly alternating between two tones indicates no continuity. For a dual deploy flight, make sure you're getting three beeps or flashes before launching! For apogee-only or motor eject flights, do what makes sense.

If idle mode is entered, you will hear an audible "di-dit" or see two short flashes ("I" for idle), and the flight state machine is disengaged, thus no ejection charges will fire. Idle mode is useful for configuring the altimeter, for extracting data from the on-board storage chip after flight, and for ground testing pyro charges.

In "Idle" and "Pad" modes, once the mode indication beeps/flashes and continuity indication has been sent, if there is no space available to log the flight in on-board memory, the flight computer will emit a warbling tone (much slower than the "no continuity tone")

See Understanding Beeps for a summary of all of the audio signals used.

Once landed, the flight computer will signal that by emitting the "Landed" sound described above, after which it will beep out the apogee height (in meters). Each digit is represented by a sequence of short "dit" beeps, with a pause between digits. A zero digit is represented with one long "dah" beep. The flight computer will continue to report landed mode and beep out the maximum height until turned off.

A.2. Ground Testing

An important aspect of preparing a rocket using electronic deployment for flight is ground testing the recovery system.

Just prep the rocket for flight, then power up the altimeter in "idle" mode. You can now command the altimeter to fire the apogee or main charges from a safe distance using your computer and the Fire Igniter tab to complete ejection testing.

A.3. Configurable Parameters

Configuring an Altus Metrum altimeter for flight is very simple. Even on our baro-only TeleMini and EasyMini boards, the use of a Kalman filter means there is no need to set a "mach delay". All of the configurable parameters can be set using AltosUI. Read Configure Altimeter for more information.

Appendix B: Handling Precautions

All Altus Metrum products are sophisticated electronic devices. When handled gently and properly installed in an air-frame, they will deliver impressive results. However, as with all electronic devices, there are some precautions you must take.

WARNING The Lithium Polymer rechargeable batteries have an extraordinary power density. This is great because we can fly with much less battery mass than if we used alkaline batteries or previous generation rechargeable batteries… but if they are punctured or their leads are allowed to short, they can and will release their energy very rapidly! Thus we recommend that you take some care when handling our batteries and consider giving them some extra protection in your air-frame. We often wrap them in suitable scraps of closed-cell packing foam before strapping them down, for example.

The barometric sensors used on all of our flight computers are sensitive to sunlight. In normal mounting situations, the baro sensor and all of the other surface mount components are "down" towards whatever the underlying mounting surface is, so this is not normally a problem. Please consider this when designing an installation in an air-frame with a see-through plastic payload bay. It is particularly important to consider this with TeleMini v1.0, both because the baro sensor is on the "top" of the board, and because many model rockets with payload bays use clear plastic for the payload bay! Replacing these with an opaque cardboard tube, painting them, or wrapping them with a layer of masking tape are all reasonable approaches to keep the sensor out of direct sunlight.

The barometric sensor sampling port must be able to "breathe", both by not being covered by foam or tape or other materials that might directly block the hole on the top of the sensor, and also by having a suitable static vent to outside air.

As with all other rocketry electronics, Altus Metrum altimeters must be protected from exposure to corrosive motor exhaust and ejection charge gasses.

Appendix C: Updating Device Firmware

EasyMini is programmed directly over their USB connectors (self programming).

TeleMini v3 can be updated directly over USB, but has no USB connector on the board. Instead, the USB signals are present on a row of 6 holes adjacent to the copyright assertion in the silk screen. Thus, updating firmware on TeleMini v3 requires making up a special cable, after which you can treat it just like TeleMetrum or TeleMega. Many USB cables seem to follow the color code of red is +5V, black is GND, green is USB +, and white is USB -. On TeleMini v3, pin 3 which has a square copper pad is ground, pin 1 is USB -, and pin 2 is USB +.

You may wish to begin by ensuring you have current firmware images. These are distributed as part of the AltOS software bundle that also includes the AltosUI ground station program. Newer ground station versions typically work fine with older firmware versions, so you don't need to update your devices just to try out new software features. You can always download the most recent version from http://www.altusmetrum.org/AltOS/

C.1. Updating EasyMini Firmware

Self-programmable devices are reprogrammed by connecting them to your computer over USB.

1. Attach a battery if necessary and power switch to the target device. Power up the device.

2. Using a Micro USB cable, connect the target device to your computer's USB socket. If the target is a TeleMini v3, make up and attach a special USB cable.

3. Run AltosUI, and select 'Flash Image' from the File menu.

4. Select the target device in the Device Selection dialog.

5. Select the image you want to flash to the device, which should have a name in the form <product>-v<product-version>-<software-version>.ihx, such as EasyMini-v1.0-1.6.0.ihx.

6. Make sure the configuration parameters are reasonable looking. If the serial number and/or RF configuration values aren't right, you'll need to change them.

7. Hit the 'OK' button and the software should proceed to flash the device with new firmware, showing a progress bar.

8. Verify that the device is working by using the 'Configure Altimeter' or 'Configure Groundstation' item to check over the configuration.

C.1.1. Recovering From Self-Flashing Failure

If the firmware loading fails, it can leave the device unable to boot. Not to worry, you can force the device to start the boot loader instead, which will let you try to flash the device again.

On each device, connecting two pins from one of the exposed connectors will force the boot loader to start, even if the regular operating system has been corrupted in some way.

EasyMini

Connect pin 6 and pin 1 of the debug connector, which is the six holes next to the beeper. Pin 1 can be identified by the square pad around it, and then the pins count sequentially across the board, making Pin 6 the one on the other end of the row.

TeleGPS v1

Connect pin 32 on the CPU to ground. Pin 32 is the right-most pin on the bottom edge of the CPU when the board is oriented with the USB and battery connectors to the right.

TeleGPS v2

Connect together pins 1 and 5 of the Debug connector. Pin 1 is the pin with the square pad around the hole.

Once you've located the right pins:

1. Turn the altimeter power off.
2. Connect a battery.
3. Connect the indicated terminals together with a short piece of wire. Take care not to accidentally connect anything else.
4. Connect USB
5. Turn the board power on.

The board should now be visible over USB as 'AltosFlash' and be ready to receive firmware. Once the board has been powered up, you can remove the piece of wire.

Appendix D: Flight Data Recording

Each flight computer logs data at 100 samples per second during ascent and 10 samples per second during descent. Data are logged to an on-board flash memory part, which can be partitioned into several equal-sized blocks, one for each flight.

Table 4. Data Storage on Altus Metrum altimeters

Device	Bytes per Sample	Total Storage	Minutes at Full Rate
EasyMini	16	1MB	10

The on-board flash is partitioned into separate flight logs, each of a fixed maximum size. Increase the maximum size of each log and you reduce the number of flights that can be stored. Decrease the size and you can store more flights.

Configuration data is also stored in the flash memory on EasyMini. This consumes 64kB of flash space. This configuration space is not available for storing flight log data.

To compute the amount of space needed for a single flight, you can multiply the expected ascent time (in seconds) by 100 times bytes-per-sample, multiply the expected descent time (in seconds) by 10 times the bytes per sample and add the two together. That will slightly under-estimate the storage (in bytes) needed for the flight.

The default size allows for several flights on each flight computer. You can adjust the size.

Altus Metrum flight computers will not overwrite existing flight data, so be sure to download flight data and erase it from the flight computer before it fills up. The flight computer will still successfully control the flight even if it cannot log data, so the only thing you will lose is the data.

Appendix E: Altus Metrum Hardware Specifications

Here's the full set of Altus Metrum products, both in production and retired.

Table 5. Altus Metrum Flight Computer Electronics

Device	Barometer	Z-axis accel	GPS	3D sensors	Storage	RF Output	Battery
EasyMini v1.0	MS5607 30km (100k')	-	-	-	1MB	-	3.7-12V
EasyMini v2.0	MS5607 30km (100k')	-	-	-	1MB	-	3.7-12V

Table 6. Altus Metrum Flight Computer Mechanical Components

Device	Connectors	Screw Terminals	Width	Length	Tube Size
EasyMini	Debug USB Battery	Apogee pyro Main pyro Battery	0.8 inch (2.03cm)	1½ inch (3.81cm)	24mm coupler

Appendix F: Release Notes

F.1. Release Notes for Version 1.9.10

Version 1.9.9

This release contains a couple of bug fixes for ground station software.

F.1.1. AltosUI

- Rework the windows DLL build to make AltosUI run on more instances of Windows 10.

F.1.2. AltosDroid

- Adapt to Android security changes which prevent AltosDroid from storing flights in /storage/emulated/0/AltusMetrum. Now, flights are stored in /storage/emulated/0/media/org.altusmetrum.AltosDroid/AltusMetrum instead. Also, AltosDroid will display an error message if flight data cannot be logged.

F.2. Release Notes for Version 1.9.9

Version 1.9.9

This release contains a critical bug fix for a problem introduced in version 1.9.8 for TeleMega and EasyMega boards. This problem occurs when using the stored configuration from 1.9.7 or earlier.

If you are running 1.9.8 or are upgrading from 1.9.8 on any version of TeleMega or EasyMega, you must reconfigure all pyro channels, recalibrate accelerometers, reset the APRS interval, adjust the beep tone and reset the pyro time.

F.2.1. AltOS

- Fix EasyMega and TeleMega upgrade process from 1.9.7 or earlier. 1.9.8 introduced larger delay values, which required modifying the configuration in-place, and the 1.9.8 version had a flaw which broke the pyro channel config and all of the config values beyond that in memory, including APRS interval, IMU accel calibation, beep tone and pyro time.

- Fix TeleMega v5.0 mag sensor driver. This driver was quite broken due to developing it in the presence of the magnetic beeper on the board. Because of that beeper, the values this sensor records are not accurate. Fortunately, they are not used for controlling the flight.

F.2.2. AltosUI

- Parse TeleMega v5.0 log files. A missing check in the code meant that the TeleMega v5.0 log files would cause an error when attempting to load them. Logs saved with AltosUI 1.9.8 were not affected, only the presentation of the data was broken.

F.3. Release Notes for Version 1.9.8

Version 1.9.8

F.3.1. AltOS

- Add support for TeleMega v5.0
- Extend extra pyro channel times to support delay > 327 seconds

F.3.2. AltosUI

- Support ARM devices in Linux binary release
- Add support for TeleMega v5.0

F.3.3. AltosDroid

- Show tilt angle in pad and flight tabs
- Show altitude as well as height (useful for TeleGPS)
- Support devices without GPS receivers

F.3.4. MicroPeak GUI

- Show error dialog if device open fails

F.4. Release Notes for Version 1.9.7

Version 1.9.7

F.4.1. AltOS

- Fix TeleGPS logging so that new data are appended to an existing log correctly

F.4.2. AltosUI

- Support Mac OS X 11 (Big Sur)
- Support Monitor Idle on Easy Timer
- Fix TeleMega v4.0 and TeleMetrum v3.0 configuration in Antenna Down mode
- Show launch sites in Load Maps view
- Add IMU header names to CSV files
- Clean up TeleGPS log corruption due to firmware bugs during firmware update

F.4.3. AltosDroid

- Support older devices back to Android version 5.1
- Fix a number of issues that could result in app crashes

F.5. Release Notes for Version 1.9.6

Version 1.9.6

F.5.1. AltOS

- Fix EasyTimer bug where it might mis-detect boost (either detect it early or not at all) due to small errors in accelerometer calibration leading to large accumulated error in speed.

- Adjust self-test of new 9-axis IMU (BMX-160) so that it doesn't think the part has a failure when tested sitting horizontally.

F.6. Release Notes for Version 1.9.5

Version 1.9.5

F.6.1. AltOS

- Add Booster mode for all dual-deploy altimeters. Select "Separation & Apogee" mode in AltosUI to fire the 'main' channel at first motor burnout and the 'apogee' channel at apogee.

- Improve reliability of stm32l firmware under heavy load. This was found using Monitor Idle on TeleMega with all flash slots full of data which would occasionally lock up.

- Fix orientation label in AltosUI for devices with radios. This makes the orientation say 'Antenna up' and 'Antenna Down' for TeleMetrum and TeleMega again.

- Fix log data for pyro channels which were inhibited (often because of a tilt limit). They used to report 'fired' as soon as they were inhibited, which was misleading. Now they never report as having been fired.

- Allow Igniter Test to fire each igniter more than once.

F.6.2. AltosUI

- Improve performance of Monitor Idle mode by sending less data for each update.

- Improve Mac OS X install scripts so that they work on Catalina. This involves sending the user to get Java from the AdoptOpenJDK project which provides a version that works with Java applications.

- Make larger dialogs scrollable for use on smaller screens.

- Fix troubles re-flashing EasyMega boards running older firmware.

- Add TeleMetrum v3.0 firmware to the Windows AltosUI packages.

F.7. Release Notes for Version 1.9.4

Version 1.9.4

F.7.1. AltOS

- Fixed firmware for TeleDongle v3.0 and TeleBT v4.0. Version 1.9.3 bricked those devices requiring them to be opened up to get them working again.

- Fixed firmware for EasyMini v1.0. Changes in the storage code caused it to use significantly more RAM which made the interrupt stack smash into other data structures and cause the device to panic at startup time.

F.7.2. AltosUI

- Add TeleMetrum v3.0 firmware to the Linux and Mac OSX AltosUI packages.

F.8. Release Notes for Version 1.9.3

Version 1.9.3

F.8.1. AltOS

- Add APRS offset. Allows multiple APRS transmitters to coordinate when transmission occurs to allow them to share a frequency.
- Fix max log size. Flight computer storage is erased in 64kB chunks. Adjust max log size to be a multiple of this size.
- Check flight erasing more carefully. Handle interrupting erasing in the middle.
- Add EasyTimer support.

F.8.2. AltosUI, TeleGPS, MicroPeak

- Add configuration support for APRS offset.
- Adjust flight log sizes to be a multiple of the flight computer erase block size.
- Report barometric pad altitude in Pad tab for TeleMini.

F.9. Release Notes for Version 1.9.1

Version 1.9.1

F.9.1. AltOS

- Add support for TeleMetrum v3.0
- Fix accel cal value when changing EasyMega v2.0 pad orientation
- Correct EasyMega v2.0 magnetometer data axes
- Fix EasyMega v2.0 idle monitor display of IMU data
- Report all sensor failures at power-up time.

F.9.2. AltosUI, TeleGPS, MicroPeak

- Display error message when attempting to graph unknown format file.
- Make it possible to disable APRS once enabled.
- Display some data for point near cursor in map tab.
- Support upgrading devices from pre-1.8 firmware
- Wait for Windows to prepare new devices during firmware upgrade. This should avoid the "COMxx: open failed" message.

F.10. Release Notes for Version 1.9

Version 1.9

F.10.1. AltOS

- Add support for EasyMega v2.0
- Replace C library for ARM devices
- Remove support for 8051-based devices and SDCC compiler

F.10.2. AltosUI, TeleGPS, MicroPeak

- Select ROM images based on product name when reflashing
- Fix TeleGPS v2 battery voltage display

F.11. Release Notes for Version 1.8.7

Version 1.8.7

F.11.1. AltOS

- Include TeleMega v3.0 firmware

F.11.2. AltosUI, TeleGPS, MicroPeak

- Poll for new devices while Device dialog is displayed
- Wait for device to re-appear when flashing new firmware
- Fetch correct TeleBT v4.0 RF calibration values from web site when reflashing.
- Change gyro headings in .csv files from x/y/z to roll/pitch/yaw
- Add documentation about Packet Link mode
- Add documentation about forcing TeleMini RF parameters to known values.
- Create a proxy server for Google Maps to re-enable map images
- Fix Java version info in all distributed jar files so that applications will run with standard Mac OS X Java.
- Replace JavaApplicationStub for Mac OS X so that applications will run with Oracle Java.

F.12. Release Notes for Version 1.8.6

Version 1.8.6

F.12.1. AltOS

- Use maximum of 64 previous tilt values in pyro tilt tests
- Eliminate 100m height requirement for coast detection
- Change After Motor pyro check to be >= instead of ==
- Change Time since Boost to be Time since launch.

F.12.2. AltosUI, TeleGPS

- Clarify pyro test phrasing
- Remove ascending/descending from pyro config UI
- Fix accel calibration in Antenna Down mode
- Add mag sensor reporting for TeleMega v3 idle monitoring.
- Fix radio parameter saving at 2400 bps telmetry rate.

F.12.3. MicroPeak

- Report altimeter-recorded maximum height value

F.13. Release Notes for Version 1.8.5

Version 1.8.5 includes fixes to the ground software support for TeleBT v4, along with a few other minor updates.

F.13.1. AltOS

- Fix startup beeps that indicate sensor failures.

F.13.2. AltosUI, TeleGPS

- When updating device firmware, make sure selected firmware matches target device.
- Correct Bluetooth device matching when looking for TeleBT devices.

F.14. Release Notes for Version 1.8.4

Version 1.8.4 includes support for EasyMini version 2.0

F.14.1. AltOS

- Support for EasyMini version 2.0 hardware.

F.15. Release Notes for Version 1.8.3

Version 1.8.3 includes support for TeleMega version 3.0 along with two important flight computer fixes. This version also changes KML export data to make Tripoli Record reporting better and some updates to graph presentation and data downloading.

F.15.1. AltOS

AltOS New Features

- Support for TeleMega version 3.0 hardware.

AltOS Bug Fixes

- Ground testing EasyMega and TeleMega additional pyro channels could result in a sticky 'fired' status which would prevent these channels from firing on future flights.
- Corrupted flight log records could prevent future flights from capturing log data.
- Fixed saving of pyro configuration that ended with 'Descending'.

F.15.2. AltosUI and TeleGPS Applications

AltosUI New Features

- Support for TeleMega version 3.0.
- Graph lines have improved appearance to make them easier to distinguish. Markers may be placed at data points to show captured recorded data values.
- Graphing offers the ability to adjust the smoothing of computed speed and acceleration data.
- The download dialog now offers to graph new flights, checks for existing files to avoid overwriting data and reports if there are checksum errors in the downloaded data.

AltosUI/TeleGPS Bug Fixes

- Restore TeleGPS tracking behavior.
- Display flight computer call sign and serial number in Monitor Idle mode instead of ground station values.

AltosUI and TeleGPS Changes

- KML export now reports both barometric and GPS altitude data to make it more useful for Tripoli record reporting.
- CSV export now includes TeleMega/EasyMega pyro voltages and tilt angle.

F.16. Release Notes for Version 1.8.2

Version 1.8.2 includes support for TeleGPS version 2.0 along with accelerometer recalibration support in AltosUI.

1.8.2 also contains a couple of minor fixes for AltosUI when analyzing saved data files.

F.16.1. AltOS

AltOS New Features

- Support for TeleGPS version 2.0 hardware.

F.16.2. AltosUI and TeleGPS Applications

AltosUI and TeleGPS New Features

- Support for TeleGPS version 2.0.
- Accelerometer re-calibration user interface.

AltosUI and TeleGPS Bug Fixes

- Prevent some crashes when reading older saved flight data for graphing or KML export.

F.17. Release Notes for Version 1.8.1

Version 1.8.1 includes an important bug fix for Apogee Lockout operation in all flight computers. Anyone using this option must update firmware.

This release also contains a change in how flight computers with accelerometers deal with speeds around and above Mach 1. In previous versions, the flight computer would completely disregard the barometric sensor above 330m/s (around Mach 1). Now, the data from the barometric sensor is reduced in effect without ever going away entirely. This prevents early drogue deployment for flights which spend considerable time above Mach 1.

1.8.1 also contains a couple of minor fixes for AltosUI when analyzing saved data files.

F.17.1. AltOS

AltOS Bug Fixes

- Handle time value wrapping in Apogee Lockout correctly. Without this, apogee lockout would sometimes prevent any drogue charge from firing.
- Change Kalman filter on flight computers with accelerometer to continue using the barometric sensor even at high speeds to avoid unintentional drogue deployment during deceleration.

F.17.2. AltosUI and TeleGPS Applications

AltosUI New Features

- Add new 'Huge' font size to make text even bigger on high resolution monitors.

AltosUI Bug Fixes

- Prevent some crashes when reading older saved flight data for graphing or KML export.
- Load frequency preference at startup. The loading code was broken, so you'd see only the default frequencies.

F.18. Release Notes for Version 1.8

Version 1.8 includes support for our new TeleBT v4.0 ground station, updates for data analysis in our ground station software and bug fixes in in the flight software for all our boards and ground station interfaces.

F.18.1. AltOS

AltOS New Features

- Add support for TeleBT v4.0 boards.

F.18.2. AltosUI and TeleGPS Applications

AltosUI New Features

- Add support for TeleBT v4.0 hardware

- Rewrite graphing and export functions. This code now handles each data series separately so that graphs are drawn correctly. Smoothing now uses a filter that looks both forward and backwards in time to make computed speed and acceleration data more accurate.

AltosUI Bug Fixes

- Correct axis labeling of magnetic sensor in TeleMega and EasyMega. The Y and Z axes were flipped.

F.19. Release Notes for Version 1.7

Version 1.7 includes support for our new TeleMini v3.0 flight computer and bug fixes in in the flight software for all our boards and ground station interfaces.

F.19.1. AltOS

AltOS New Features

- Add support for TeleMini v3.0 boards.

AltOS Fixes

- Fix interrupt priorities on STM32L processors. Run timer interrupt at lowest priority so that device interrupts get serviced first.

F.19.2. AltosUI and TeleGPS Applications

AltosUI New Features

- Add support for TeleMini v3.0 hardware

F.20. Release Notes for Version 1.6.8

Version 1.6.8 fixes a TeleMega and TeleMetrum v2.0 bug where the device could stop logging data and transmitting telemetry in flight. All TeleMega v1.0, v2.0 and TeleMetrum v2.0 users should update their flight firmware.

F.20.1. AltOS

AltOS fixes:

- Fix STM32L DMA driver to work around STM32L SoC DMA priority issue t lock-up in the logging or radio code, either of which could stop data logging and telemetry.
- Avoid having TeleBT battery status values smash telemetry packet data by holding a lock during both operations.
- Don't start reporting 'flight 0' when the log fills in flight.

AltOS changes:

- Flash LEDS on all products briefly during power up so that they can be tested during production.

F.20.2. AltosUI, TeleGPS and AltosDroid Applications

AltosUI fixes:

- Re-enable go/no-go entries after they've been disabled due to lack of data. If telemetry information is delayed when the Ui starts up, sometimes important fields would get disabled to never re-appear.
- Deal with ground station failure better during Configure Ground Station operation by cleaning up pending operations.

F.21. Release Notes for Version 1.6.5

Version 1.6.5 fixes a TeleMega and TeleMetrum v2.0 bug where the device would often stop logging data and transmitting telemetry in flight. All TeleMega v1.0, v2.0 and TeleMetrum v2.0 users should update their flight firmware.

F.21.1. AltOS

AltOS fixes:

- Fix STM32L SPI driver to prevent lock-up in the logging or radio code, either of which could stop data logging and telemetry. Found and characterized by Chuck Haskin, who also tested the new firmware before release.

F.21.2. AltosUI, TeleGPS and AltosDroid Applications

AltosUI fixes:

- Deliver firmward for TeleMega v2.0 and TeleBT v3.0 with Windows package.

F.22. Release Notes for Version 1.6.4

Version 1.6.4 fixes a bluetooth communication problem with TeleBT v1.0 devices, along with some altosui and altosdroid minor nits. It also now ships firmware for some newer devices.

F.22.1. AltOS

AltOS fixes:

- Fix hardware flow control on TeleBT v1.0. Hardware RTS/CTS doesn't seem to work, switch from using the hardware to driving these pins with software.
- Fix ARM USB drivers to deal with OS restarts. Needed to reset all USB-related state when the USB bus is reset. These fixes affect all STM32L, STM32F0 and LPC11U14 based devices.

F.22.2. AltosUI, TeleGPS and AltosDroid Applications

AltosUI, TeleGPS and AltosDroid New Features:

- Automatically switch from meters or feet to kilometers or miles for distance units.
- Add Monitor Idle mode to TeleGPS application.

AltosUI, TeleGPS and AltosDroid Fixes:

- Abort map preloading when the preload map dialog is closed.
- In AltosDroid, Don't reconnect to last device if the user had disconnected it the last time the application was active.

F.22.3. Documentation

- Mention TeleMega v2.0 in hardware specs table.
- Document TeleGPS RF output in telegps manual.

F.23. Release Notes for Version 1.6.3

Version 1.6.3 adds idle mode to AltosDroid and has bug fixes for our host software on desktops, laptops an android devices along with BlueTooth support for Windows.

F.23.1. AltOS

AltOS fixes:

- Fix hardware flow control on TeleBT v3.0. RTS/CTS is wired backwards on this board, switch from using the hardware to driving these pins with software.

F.23.2. AltosUI and TeleGPS Applications

AltosUI and TeleGPS New Features:

- Add BlueTooth support for Windows operating system. This supports connections to TeleBT over BlueTooth rather than just USB.

AltosUI and TeleGPS Fixes:

- Change Java detection and install on Windows. Detection is now done by looking for the 'javaw.exe' program, and installation by opening a browser on the java.com web site.
- Delay polling while the Fire Igniters is visible to allow for TeleMega to report back complete status over the radio.
- Disallow changing RF calibration numbers in the configuration UI. There's no good reason to change this from the field, and recovering is really hard if you haven't written down the right number.
- Fix USB device discovery on Mac OS X El Capitan. This makes the connected Altus Metrum USB devices appear again.
- Fix acceleration data presented in MonitorIdle mode for TeleMetrum v2.0 flight computers.

F.23.3. AltosDroid

AltosDroid new features:

- Monitor Idle mode. Check state of flight computer while in idle mode over the radio link
- Fire Igniters. Remotely fire ignires for recovery system ground tests.
- Remote reboot. Cause the flight computer to reboot over the radio link. This provides a method for switching the flight computer from idle to flight mode without needing to reach the power switch.
- Configurable frequency menu. Change the set of available frequencies and provide more descriptive names.

AltosDroid bug fixes:

- Don't set target location if GPS hasn't locked yet.
- Fix saving target states so they can be reloaded when the application restarts. When the application is shut down and restarted, all previous target state information will be restored (including GPS position if available).
- Fix crash on some Android devices for offline maps when changing the map scale or location.
- Don't require USB OTG support. This kept the latest AltosDroid from being offered on devices without USB device support, although it can work without that just fine using BlueTooth.

- Don't require bluetooth to be enabled. This allows the application to operate with USB devices or just show old data without turning on the bluetooth radio.

- Recover old tracker positions when restarting application. This finally allows you to safely stop and restart the application without losing the last known location of any tracker.

F.23.4. Documentation

- Document TeleMega and EasyMega additional pyro channel continuity audio alert pattern.

F.24. Release Notes for Version 1.6.2

Version 1.6.2 includes support for our updated TeleMega v2.0 product and bug fixes in in the flight software for all our boards and ground station interfaces.

F.24.1. AltOS

AltOS New Features:

- Add support for TeleMega v2.0 boards.
- Add PWM servo driver. There's no higher level code using this yet, but the driver allows testing of the TeleMega v2.0 servo output connector.

AltOS Fixes:

- Slow down telemetry packets to allow receiver to keep up.

F.24.2. AltosUI and TeleGPS Applications

AltosUI and TeleGPS Fixes:

- Fix post-flight orientation computation when processing TeleMega and EasyMega eeprom data files.
- Capture complete eeprom data even when there are invalid entries in the data. This keeps reading eeprom contents and writing the associated .eeprom file when an error is detected.

F.24.3. Documentation

We spent a bunch of time trying to improve our documentation

- HTML versions now have a table of contents on the left side.
- EasyMini now has its own shorter manual.
- Provide links between sections in each document.
- Lots of minor rewriting and restructuring to avoid duplication of information