



# **The Altus Metrum System**

*An Owner's Manual for Altus Metrum Rocketry Electronics*

Keith Packard, Bdale Garbee, Bob Finch, Anthony Towns

Version v1.9.22.2, 13 Oct 2025

# Table of Contents

- License ..... 1
- Acknowledgments ..... 2
- 1. Introduction and Overview ..... 3
- 2. Getting Started ..... 4
  - 2.1. Batteries ..... 4
  - 2.2. Ground Station Hardware ..... 4
  - 2.3. Linux/Mac/Windows Ground Station Software ..... 5
  - 2.4. Android Ground Station Software ..... 5
- 3. Using Altus Metrum Hardware ..... 6
  - 3.1. Wiring and Electrical Interference ..... 6
  - 3.2. Hooking Up Batteries ..... 6
  - 3.3. Hooking Up Pyro Charges ..... 6
  - 3.4. Hooking Up a Power Switch ..... 7
  - 3.5. Understanding Beeps ..... 7
  - 3.6. Turning On the Power ..... 8
  - 3.7. Using an External Active Switch Circuit ..... 9
  - 3.8. Using a Separate Pyro Battery ..... 9
  - 3.9. Using Packet Link Mode ..... 9
- 4. TeleMetrum ..... 10
  - 4.1. TeleMetrum Screw Terminals ..... 10
  - 4.2. Using a Separate Pyro Battery with TeleMetrum ..... 11
  - 4.3. Using an Active Switch with TeleMetrum ..... 11
- 5. TeleMini ..... 12
  - 5.1. TeleMini v3 Screw Terminals ..... 12
  - 5.2. Using a Separate Pyro Battery with TeleMini v3 ..... 13
  - 5.3. Using an Active Switch with TeleMini v3 ..... 13
  - 5.4. Using Packet Link Mode with TeleMini ..... 13
  - 5.5. Forcing TeleMini radio parameters to known defaults ..... 14
  - 5.6. TeleMini v1 ..... 14
- 6. EasyMini ..... 15
  - 6.1. EasyMini Screw Terminals ..... 15
  - 6.2. Connecting A Battery To EasyMini ..... 15
  - 6.3. Charging Lithium Batteries ..... 16
  - 6.4. Using a Separate Pyro Battery with EasyMini ..... 16
  - 6.5. Using an Active Switch with EasyMini ..... 16
- 7. TeleMega ..... 17
  - 7.1. TeleMega Screw Terminals ..... 17
  - 7.2. Using a Separate Pyro Battery with TeleMega ..... 18

- 7.3. Using Only One Battery With TeleMega . . . . . 18
- 7.4. Using an Active Switch with TeleMega . . . . . 18
- 8. EasyMega . . . . . 19
  - 8.1. EasyMega Screw Terminals . . . . . 19
  - 8.2. Using a Separate Pyro Battery with EasyMega . . . . . 20
  - 8.3. Using Only One Battery With EasyMega . . . . . 20
  - 8.4. Using an Active Switch with EasyMega . . . . . 20
- 9. EasyTimer . . . . . 21
  - 9.1. EasyTimer Screw Terminals . . . . . 21
  - 9.2. Connecting A Battery To EasyTimer . . . . . 21
  - 9.3. Charging Lithium Batteries . . . . . 22
  - 9.4. Using a Separate Pyro Battery with EasyTimer . . . . . 22
  - 9.5. Using an Active Switch with EasyTimer . . . . . 22
- 10. Installation . . . . . 23
- 11. Using Altus Metrum Products . . . . . 24
  - 11.1. Being Legal . . . . . 24
  - 11.2. In the Rocket . . . . . 24
  - 11.3. On the Ground . . . . . 24
  - 11.4. Data Analysis . . . . . 25
  - 11.5. Future Plans . . . . . 25
- 12. AltosUI . . . . . 27
  - 12.1. Monitor Flight . . . . . 27
  - 12.2. Save Flight Data . . . . . 33
  - 12.3. Replay Flight . . . . . 34
  - 12.4. Graph Data . . . . . 34
  - 12.5. Export Data . . . . . 37
  - 12.6. Configure Altimeter . . . . . 37
  - 12.7. Configure AltosUI . . . . . 45
  - 12.8. Configure Groundstation . . . . . 47
  - 12.9. Flash Image . . . . . 48
  - 12.10. Fire Igniter . . . . . 48
  - 12.11. Scan Channels . . . . . 49
  - 12.12. Load Maps . . . . . 49
  - 12.13. Monitor Idle . . . . . 50
- 13. AltosDroid . . . . . 52
  - 13.1. Installing AltosDroid . . . . . 52
  - 13.2. Charging TeleBT Battery . . . . . 52
  - 13.3. Connecting to TeleBT over Bluetooth™ . . . . . 52
  - 13.4. Connecting to TeleDongle or TeleBT over USB . . . . . 52
  - 13.5. AltosDroid Menu . . . . . 53

13.6. Setup	53
13.7. Idle Mode	54
13.8. AltosDroid Flight Monitoring	54
13.9. Pad	54
13.10. Flight	56
13.11. Recover	57
13.12. Map	57
13.13. Downloading Flight Logs	58
Appendix A: System Operation	59
A.1. Firmware Modes	59
A.2. GPS	60
A.3. Controlling An Altimeter Over The Radio Link	60
A.4. Ground Testing	61
A.5. Radio Link	62
A.6. APRS	62
A.7. Configurable Parameters	63
Appendix B: Example Pyro Channel Configurations	65
B.1. Two-Stage Flights	65
B.2. Triggered Clusters and Air Starts	65
B.3. Redundant Apogee	66
B.4. Redundant Main	66
B.5. Apogee Above Baro Sensor Limit	66
Appendix C: Handling Precautions	69
Appendix D: Updating Device Firmware	70
D.1. Updating TeleMega, TeleMetrum v2 or newer, TeleMini v3, EasyMega, EasyMini, TeleDongle v3 or TeleBT v3 Firmware	70
D.2. Pair Programming	72
Appendix E: Frequency Calibration	75
E.1. Background Information	75
E.2. Required Equipment	75
E.3. RF Calibration Procedure	75
Appendix F: Flight Data Recording	77
Appendix G: Hardware Specifications	78
Appendix H: Release Notes	81
H.1. Release Notes for Version 1.9.22	81
H.2. Release Notes for Version 1.9.21	82
H.3. Release Notes for Version 1.9.20	83
H.4. Release Notes for Version 1.9.19	84
H.5. Release Notes for Version 1.9.18	85
H.6. Release Notes for Version 1.9.17	86

H.7. Release Notes for Version 1.9.16 .....	87
H.8. Release Notes for Version 1.9.15 .....	88
H.9. Release Notes for Version 1.9.14 .....	89
H.10. Release Notes for Version 1.9.13 .....	90
H.11. Release Notes for Version 1.9.12 .....	91
H.12. Release Notes for Version 1.9.11 .....	92
H.13. Release Notes for Version 1.9.10 .....	93
H.14. Release Notes for Version 1.9.9 .....	94
H.15. Release Notes for Version 1.9.8 .....	95
H.16. Release Notes for Version 1.9.7 .....	96
H.17. Release Notes for Version 1.9.6 .....	97
H.18. Release Notes for Version 1.9.5 .....	98
H.19. Release Notes for Version 1.9.4 .....	99
H.20. Release Notes for Version 1.9.3 .....	100
H.21. Release Notes for Version 1.9.2 .....	101
H.22. Release Notes for Version 1.9.1 .....	102
H.23. Release Notes for Version 1.9 .....	103
H.24. Release Notes for Version 1.8.7 .....	104
H.25. Release Notes for Version 1.8.6 .....	105
H.26. Release Notes for Version 1.8.5 .....	106
H.27. Release Notes for Version 1.8.4 .....	107
H.28. Release Notes for Version 1.8.3 .....	108
H.29. Release Notes for Version 1.8.2 .....	109
H.30. Release Notes for Version 1.8.1 .....	110
H.31. Release Notes for Version 1.8 .....	111
H.32. Release Notes for Version 1.7 .....	112
H.33. Release Notes for Version 1.6.8 .....	113
H.34. Release Notes for Version 1.6.5 .....	114
H.35. Release Notes for Version 1.6.4 .....	115
H.36. Release Notes for Version 1.6.3 .....	116
H.37. Release Notes for Version 1.6.2 .....	118
H.38. Release Notes for Version 1.6.1 .....	119
H.39. Release Notes for Version 1.6 .....	121
H.40. Release Notes for Version 1.5 .....	123
H.41. Release Notes for Version 1.4.2 .....	125
H.42. Release Notes for Version 1.4.1 .....	126
H.43. Release Notes for Version 1.4 .....	127
H.44. Release Notes for Version 1.3.2 .....	129
H.45. Release Notes for Version 1.3.1 .....	130
H.46. Release Notes for Version 1.3 .....	131

H.47. Release Notes for Version 1.2.1 .....	132
H.48. Release Notes for Version 1.2 .....	134
H.49. Release Notes for Version 1.1 .....	135
H.50. Release Notes for Version 1.1 .....	136
H.51. Release Notes for Version 1.0.1 .....	138
H.52. Release Notes for Version 0.9.2 .....	140
H.53. Release Notes for Version 0.9 .....	141
H.54. Release Notes for Version 0.8 .....	142
H.55. Release Notes for Version 0.7.1 .....	143

# License

Copyright © 2025 Bdale Garbee and Keith Packard

This document is released under the terms of the [Creative Commons ShareAlike 3.0 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, K7WQ  
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://groups.io/g/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

For TeleMetrum, TeleMega and EasyMega, the battery can be charged by plugging it into the corresponding socket of the device and then using the USB cable to plug the flight computer into your computer's USB socket. The on-board circuitry will charge the battery whenever it is plugged in, because the on-off switch does NOT control the charging circuitry. The Lithium Polymer EasyTimer, TeleMini and 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 EasyTimer and 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.



On TeleMetrum v1 boards, when the GPS chip is initially searching for satellites, TeleMetrum will consume more current than it pulls from the USB port, so the battery must be attached in order to get satellite lock. Once GPS is locked, the current consumption goes back down enough to enable charging while running. So it's a good idea to fully charge the battery as your first item of business so there is no issue getting and maintaining satellite lock. The yellow charge indicator led will go out when the battery is nearly full and the charger goes to trickle charge. It can take several hours to fully recharge a deeply discharged battery.

TeleMetrum v2 or newer, TeleMega and EasyMega use a higher power battery charger, allowing them to charge the battery while running the board at maximum power. When the battery is charging, or when the board is consuming a lot of power, the red LED will be lit. When the battery is fully charged, the green LED will be lit. When the battery is damaged or missing, both LEDs will be lit, which appears yellow.

## 2.2. Ground Station Hardware

There are two ground stations available, the TeleDongle USB to RF interface and the TeleBT Bluetooth/USB to RF interface. If you plug either of these in to your Mac or Linux computer it should "just work", showing up as a serial port device. Windows systems need driver information that is part of the AltOS download to know that the existing USB modem driver will work. We therefore recommend installing our software before plugging in TeleDongle if you are using a Windows computer. If you are using an older version of Linux and are having problems, try moving to a fresher kernel (2.6.33 or newer).

## 2.3. 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>

## 2.4. Android Ground Station Software

TeleBT can also connect to an Android device over BlueTooth or USB. The [AltosDroid Android application](#) is available from the [Google Play system](#).

You don't need a data plan to use AltosDroid, but without network access, you'll want to download offline map data before wandering away from the network.

# 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 Batteries

All Altus Metrum flight computers have a two pin JST PH series connector intended for connection of a single-cell Lithium Polymer (LiPo) cell (3.7V nominal). We recommend using the batteries sold by Altus Metrum.

You can purchase 1S LiPo batteries from other vendors, but note the warnings below. If you are prohibited from using lithium polymer batteries for some reason, we believe 3.7 volt nominal cylindrical lithium ion cells in 14500 or 18650 form will work, but have not tested this ourselves. To avoid problems during pyro events, use unprotected cells with our flight computers. Protected cells may be used with non-pyro devices like TeleGPS or EasyMotor.

If you're not using batteries sold by Altus Metrum, on the white JST connector, pin 1 is positive (square pad), pin 2 is negative. Spark Fun sells a suitable cable with the connector pre-attached (crimping pins for these connectors is hard to do well, we recommend letting someone professional do it!), which they call a [JST Jumper 2 Wire Assembly](#)



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.



Most LiPo batteries from sources other than Altus Metrum have protection circuit boards, the behavior of which may cause power to be completely shut off during a pyro event and not come back on until the board is power-cycled after flight. See the How-To Documents at [Misc Documents](#) for details on how to remove such protection boards.

Products that include a battery charger employ a constant-current / constant-voltage charge algorithm with preconditioning, charge termination, and thermal protection. The constant voltage phase is fixed at 4.2V, the constant current phase setting varies by product version, and can be found in [Hardware Specifications](#).

## 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. TeleMini doesn't have room for a beeper, so instead it uses an LED, which works the same, except for every beep is replaced with the flash of the LED.

Here's a short summary of all of the modes and the beeping (or flashing, in the case of TeleMini v1) 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 or radio link.
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.

Name	Beeps	Description
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.
Additional Igniters	four very short beeps	Continuity indication for the additional pyro channels on TeleMega, EasyMega and EasyTimer. One high tone for no continuity, one low tone for continuity. On TeleMega and EasyMega, these are produced after the continuity indicators for the two primary igniter channels.

For devices with a radio transmitter, in addition to the digital and APRS telemetry signals, you can also receive audio tones with a standard amateur 70cm FM receiver. While on the pad, you will hear igniter status once every five seconds.

Table 3. Pad Radio Indications

Name	Beeps	Description
Neither	½ second tone	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.

During ascent, the tones will be muted to allow the telemetry data to consume the full radio bandwidth.

During descent and after landing, a ½ second tone will be transmitted every five seconds. This can be used to find the rocket using RDF techniques when the signal is too weak to receive GPS information via telemetry or APRS.

## 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, and the radio goes into transmit-only mode. 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 and in packet mode over the radio. 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 flight computers with accelerometers (TeleMetrum, EasyMega, TeleMega and EasyTimer), the mode is selected by the orientation of the board during the self test interval. If the board is

pointing upwards as if ready to fly, it will enter Flight/Pad mode. Otherwise, it will enter Idle mode.

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

For TeleMini v1.0, if a packet link is waiting to connect when the device is powered on, 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 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. TeleMetrum



Figure 1. TeleMetrum v2 Board

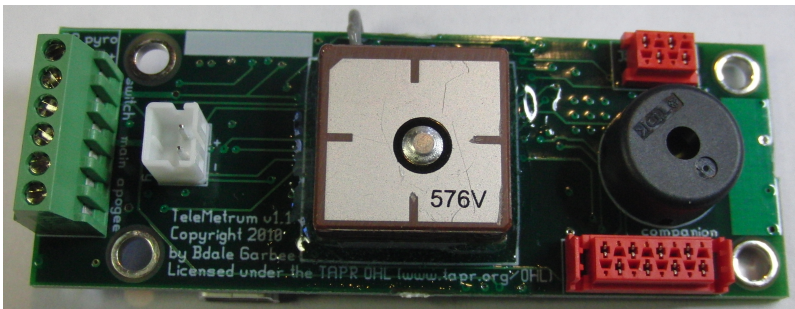


Figure 2. TeleMetrum v1 Board

TeleMetrum is a 1 inch by 2¾ inch circuit board. It was designed to fit inside coupler for 29mm air-frame tubing, but using it in a tube that small in diameter may require some creativity in mounting and wiring to succeed! The presence of an accelerometer means TeleMetrum should be aligned along the flight axis of the airframe, and by default the ¼ wave UHF wire antenna should be on the nose-cone end of the board. The antenna wire is about 7 inches long, and wiring for a power switch and the e-matches for apogee and main ejection charges depart from the fin can end of the board, meaning an ideal “simple” avionics bay for TeleMetrum should have at least 10 inches of interior length.

There are two generations of the TeleMetrum design. The major changes after v1 generation are:

- uBlox GPS chip certified for altitude records
- Higher power radio (40mW vs 10mW)
- APRS support

Otherwise, they’re the same size, with mounting holes and screw terminals in the same position.

## 4.1. TeleMetrum Screw Terminals

TeleMetrum has six screw terminals on the end of the board opposite the telemetry antenna. Two are for the power switch, and two each for the apogee and main igniter circuits. Using the picture above and starting from the top, the terminals are as follows:

Table 4. TeleMetrum Screw Terminals

Terminal #	Terminal Name	Description
1	Switch Output	Switch connection to flight computer
2	Switch Input	Switch connection to positive battery terminal
3	Main +	Main pyro channel common connection to battery +
4	Main -	Main pyro channel connection to pyro circuit
5	Apogee +	Apogee pyro channel common connection to battery +
6	Apogee -	Apogee pyro channel connection to pyro circuit

## 4.2. Using a Separate Pyro Battery with TeleMetrum

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 battery terminal to the TeleMetrum ground, insert a small piece of wire, 24 to 28 gauge stranded, into the GND hole just above the screw terminal strip and solder it in place.

Connecting the switched positive battery terminal to the pyro charges must be done separate from TeleMetrum, by soldering them together or using some other connector. Note that you must include a switch in the positive lead from the pyro battery for safety!

The other lead from each pyro charge is then inserted into the appropriate per-pyro channel screw terminal (terminal 4 for the Main charge, terminal 6 for the Apogee charge).

## 4.3. Using an Active Switch with TeleMetrum

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.

The positive battery terminal is available on screw terminal 2, the positive flight computer input is on terminal 1. To hook a lead to ground, solder a piece of wire, 24 to 28 gauge stranded, to the GND hole just above terminal 1.

# Chapter 5. TeleMini

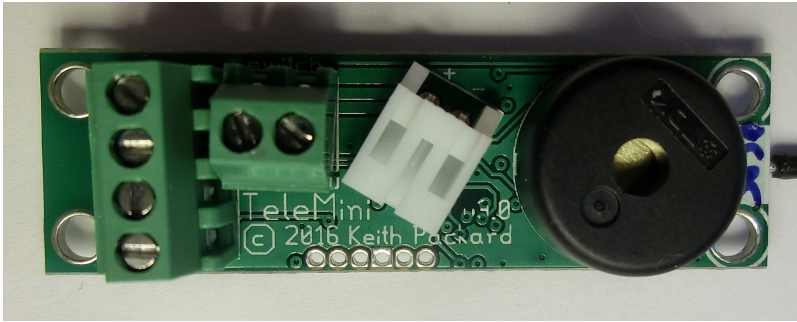
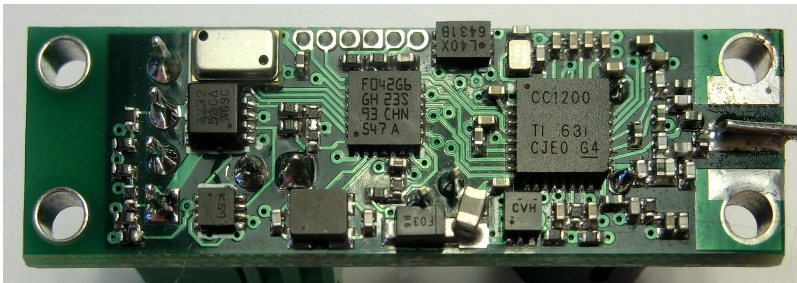


Figure 3. TeleMini v3 Board



TeleMini v3 is 0.5 inches by 1.67 inches. It was designed to fit inside an 18mm air-frame tube, but using it in a tube that small in diameter may require some creativity in mounting and wiring to succeed! Since there is no accelerometer, TeleMini can be mounted in any convenient orientation. The default  $\frac{1}{4}$  wave UHF wire antenna attached to the center of one end of the board is about 7 inches long. Screw terminals for the power switch are located in the middle of the board. Screw terminals for the e-matches for apogee and main ejection charges depart from the other end of the board, meaning an ideal “simple” avionics bay for TeleMini should have at least 9 inches of interior length.

## 5.1. TeleMini v3 Screw Terminals

TeleMini v3 has four screw terminals on the end of the board opposite the telemetry antenna. Two are for the apogee and two are for main igniter circuits. Another two screw terminals are located in the middle of the board for the power switch. Using the picture above and starting from the top for the pyro terminals and from the left for the power switch terminals, the connections are as follows:

Table 5. TeleMini v3 Screw Terminals

Terminal #	Terminal Name	Description
1	Apogee -	Apogee pyro channel connection to pyro circuit
2	Apogee +	Apogee pyro channel common connection to battery +
3	Main -	Main pyro channel connection to pyro circuit
4	Main +	Main pyro channel common connection to battery +
Left	Switch Output	Switch connection to flight computer
Right	Switch Input	Switch connection to positive battery terminal

## 5.2. Using a Separate Pyro Battery with TeleMini v3

As described above, using an external pyro battery involves connecting the negative battery terminal to the flight computer ground, connecting the switched positive battery terminal to one of the igniter leads and connecting the other igniter lead to the per-channel pyro circuit connection. Because there is no solid ground connection to use on TeleMini, this is not recommended.

The only available ground connection on TeleMini v3 are the two mounting holes next to the telemetry antenna. Somehow connect a small piece of wire to one of those holes and hook it to the negative pyro battery terminal.

Connecting the positive battery terminal to the pyro charges must be done separate from TeleMini v3, by soldering them together or using some other connector. Note that you must include a switch in the pyro battery positive lead for safety!

The other lead from each pyro charge is then inserted into the appropriate per-pyro channel screw terminal (terminal 3 for the Main charge, terminal 1 for the Apogee charge).

## 5.3. Using an Active Switch with TeleMini v3

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. Again, because TeleMini doesn't have any good ground connection, this is not recommended.

The positive battery terminal is available on the Right power switch wire, the positive flight computer input is on the left power switch wire. Hook a lead to either of the mounting holes for a ground connection.

## 5.4. Using Packet Link Mode with TeleMini

After TeleMini powers up, it will check to see if some device is attempting to communicate with it using Packet Link Mode. If so, it will switch to idle mode and start communicating. To switch to flight mode, reboot the device either over the radio link or by powering it off and back on.

If no ground station is attempting to communicate using Packet Link Mode, TeleMini will enter pad mode and prepare for flight.

The sequence of operations to use Packet Link Mode with TeleMini is:

1. Configure the ground station data rate, frequency and callsign to match the TeleMini settings.
2. Start Packet Link Mode in the ground station by selecting the desired operation (Safe Flight Data, Configure Altimeter, Fire Igniter or Monitor Idle). Select the TeleBT or TeleDongle device. The red LED should begin flashing rapidly.
3. Turn on TeleMini. You should see the red LED flash very rapidly during the initial communication burst, but it should then slow down when the link is idle.

Once TeleMini is in Idle mode, it will stay in that mode until rebooted. That means you can stop one Packet Link operation, wait a while and start another Packet Link operation.

## 5.5. Forcing TeleMini radio parameters to known defaults

If you don't know what the TeleMini frequency and callsign settings are, you can temporarily force it back to the original default values (frequency 434.550MHz, callsign NOCALL) by connecting a wire between hole 3 and hole 6 on the debug connector. Hole 3 has the square pad around it, hole 6 is the one nearest the MS5607 baro sensor, which is a rectangular component with a metal cap that has two holes in it.

Once TeleMini has been powered up with this wire connected, the wire may be removed. The radio parameters will stay set to these default values until changed by the user or when the device is rebooted.

## 5.6. TeleMini v1

TeleMini v1 is the earlier version of this product. It has a lower-power radio, less storage, no beeper and soldered-in wires instead of screw terminals for the power switch.

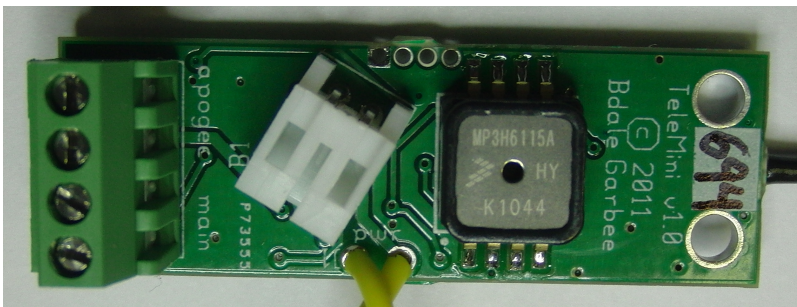


Figure 4. TeleMini v1 Board

# Chapter 6. EasyMini

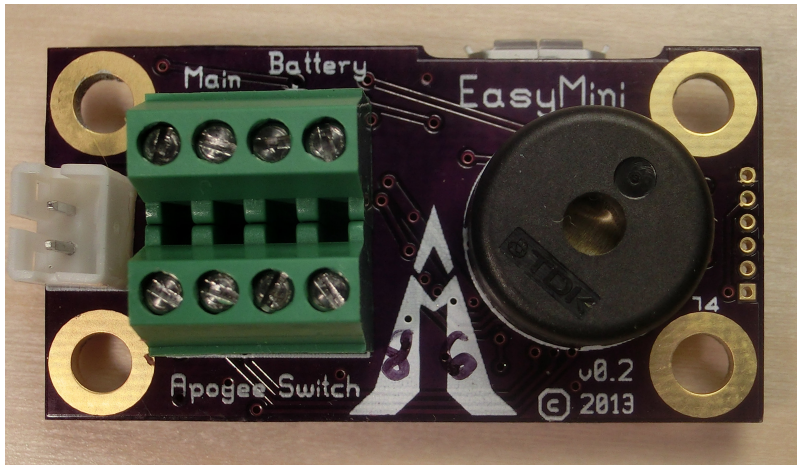


Figure 5. EasyMini Board

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.

## 6.1. 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 6. 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

## 6.2. 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.

### 6.3. 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 Lithium Polymer cell, you'll need an external charger. These are available from Altus Metrum, or from Spark Fun.

### 6.4. 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).

### 6.5. 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 7. TeleMega

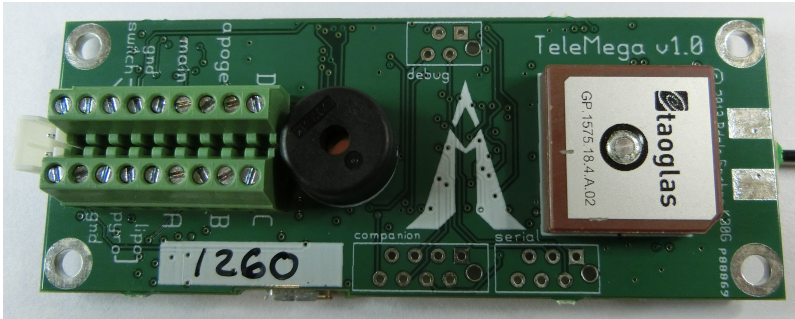


Figure 6. TeleMega Board

TeleMega is a 1¼ inch by 3¼ inch circuit board. It was designed to easily fit in a 38mm coupler. Like TeleMetrum, TeleMega has an accelerometer and so it must be mounted so that the board is aligned with the flight axis. It can be mounted either antenna up or down.

TeleMega v2.0 has a few minor changes from v1.0:

- Companion connector matches EasyMega functions
- Serial port connector replaced with servo connector with support for up to 4 PWM channels.
- Radio switched from cc1120 to cc1200.

None of these affect operation using the stock firmware, but they do mean that the device needs different firmware to operate correctly, so make sure you load the right firmware when reflashing the device.

## 7.1. TeleMega Screw Terminals

TeleMega has two sets of nine screw terminals on the end of the board opposite the telemetry antenna. They are as follows:

Table 7. TeleMega Screw Terminals

Terminal #	Terminal Name	Description
Top 1	Switch Input	Switch connection to positive battery terminal
Top 2	Switch Output	Switch connection to flight computer
Top 3	GND	Ground connection for use with external active switch
Top 4	Main -	Main pyro channel connection to pyro circuit
Top 5	Main +	Main pyro channel common connection to battery +
Top 6	Apogee -	Apogee pyro channel connection to pyro circuit
Top 7	Apogee +	Apogee pyro channel common connection to battery +
Top 8	D -	D pyro channel connection to pyro circuit
Top 9	D +	D pyro channel common connection to battery +
Bottom 1	GND	Ground connection for negative pyro battery terminal
Bottom 2	Pyro	Positive pyro battery terminal
Bottom 3	Lipo	Power switch output. Use to connect main battery to pyro battery input
Bottom 4	A -	A pyro channel connection to pyro circuit

Terminal #	Terminal Name	Description
Bottom 5	A +	A pyro channel common connection to battery +
Bottom 6	B -	B pyro channel connection to pyro circuit
Bottom 7	B +	B pyro channel common connection to battery +
Bottom 8	C -	C pyro channel connection to pyro circuit
Bottom 9	C +	C pyro channel common connection to battery +

## 7.2. Using a Separate Pyro Battery with TeleMega

TeleMega provides explicit support for an external pyro battery. All that is required is to remove the jumper between the lipo terminal (Bottom 3) and the pyro terminal (Bottom 2). Then hook the negative pyro battery terminal to ground (Bottom 1) and the switched positive pyro battery to the pyro battery input (Bottom 2). Note that you must include a switch between the pyro battery and the board for safety, as the on-board power switch circuit only supports the primary battery! You can then use the existing pyro screw terminals to hook up all of the pyro charges.

## 7.3. Using Only One Battery With TeleMega

Because TeleMega has built-in support for a separate pyro battery, if you want to fly with just one battery running both the computer and firing the charges, you need to connect the flight computer battery to the pyro circuit. TeleMega has two screw terminals for this—hook a wire from the Lipo terminal (Bottom 3) to the Pyro terminal (Bottom 2).

## 7.4. Using an Active Switch with TeleMega

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.

The positive battery terminal is available on Top terminal 1, the positive flight computer input is on Top terminal 2. Ground is on Top terminal 3.

# Chapter 8. EasyMega



Figure 7. EasyMega Board

EasyMega is a 1¼ inch by 2¼ inch circuit board. It was designed to easily fit in a 38mm coupler. Like TeleMetrum, EasyMega has an accelerometer and so it must be mounted so that the board is aligned with the flight axis. It can be mounted either antenna up or down.

## 8.1. EasyMega Screw Terminals

EasyMega has two sets of nine screw terminals on the end of the board opposite the telemetry antenna. They are as follows:

Table 8. EasyMega Screw Terminals

Terminal #	Terminal Name	Description
Top 1	Switch Input	Switch connection to positive battery terminal
Top 2	Switch Output	Switch connection to flight computer
Top 3	GND	Ground connection for use with external active switch
Top 4	Main -	Main pyro channel connection to pyro circuit
Top 5	Main +	Main pyro channel common connection to battery +
Top 6	Apogee -	Apogee pyro channel connection to pyro circuit
Top 7	Apogee +	Apogee pyro channel common connection to battery +
Top 8	D -	D pyro channel connection to pyro circuit
Top 9	D +	D pyro channel common connection to battery +
Bottom 1	GND	Ground connection for negative pyro battery terminal
Bottom 2	Pyro	Positive pyro battery terminal
Bottom 3	Lipo	Power switch output. Use to connect main battery to pyro battery input
Bottom 4	A -	A pyro channel connection to pyro circuit
Bottom 5	A +	A pyro channel common connection to battery +
Bottom 6	B -	B pyro channel connection to pyro circuit
Bottom 7	B +	B pyro channel common connection to battery +
Bottom 8	C -	C pyro channel connection to pyro circuit
Bottom 9	C +	C pyro channel common connection to battery +

## 8.2. Using a Separate Pyro Battery with EasyMega

EasyMega provides explicit support for an external pyro battery. All that is required is to remove the jumper between the lipo terminal (Bottom 3) and the pyro terminal (Bottom 2). Then hook the negative pyro battery terminal to ground (Bottom 1) and the switched positive pyro battery to the pyro battery input (Bottom 2). Note that you must include a switch between the pyro battery and the board for safety, as the on-board power switch circuit only supports the primary battery! You can then use the existing pyro screw terminals to hook up all of the pyro charges.

## 8.3. Using Only One Battery With EasyMega

Because EasyMega has built-in support for a separate pyro battery, if you want to fly with just one battery running both the computer and firing the charges, you need to connect the flight computer battery to the pyro circuit. EasyMega has two screw terminals for this—hook a wire from the Lipo terminal (Bottom 3) to the Pyro terminal (Bottom 2).

## 8.4. Using an Active Switch with EasyMega

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.

The positive battery terminal is available on Top terminal 1, the positive flight computer input is on Top terminal 2. Ground is on Top terminal 3.

# Chapter 9. EasyTimer

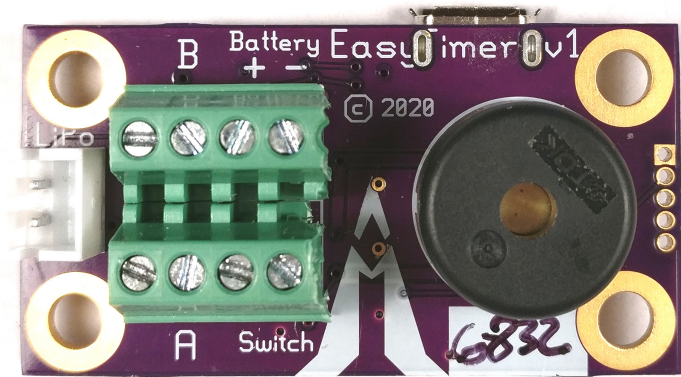


Figure 8. EasyTimer Board

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

EasyTimer is designed to control events during ascent. It has an accelerometer and gyroscope that can measure acceleration and rotation and compute speed and tilt angle. EasyTimer has two pyro channels which can be configured to fire at various points during flight. Because EasyTimer has no barometric sensor, it cannot be used to fire recovery charges at apogee or during descent. EasyTimer is configured using the AltosUI application which is available for Linux, Mac OS X and Windows.

## 9.1. EasyTimer Screw Terminals

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

Table 9. EasyTimer Screw Terminals

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

## 9.2. Connecting A Battery To EasyTimer

There are two possible battery connections on EasyTimer. 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.

### 9.3. Charging Lithium Batteries

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

### 9.4. Using a Separate Pyro Battery with EasyTimer

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 EasyTimer 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 EasyTimer, 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 pyro channel A charge, bottom terminal 1 for pyro channel B charge).

### 9.5. Using an Active Switch with EasyTimer

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 10. 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.



Check polarity and voltage before connecting any battery not purchased from Altus Metrum.



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.

Except for TeleMini v1.0, the flight computers also use the screw terminal block for the power switch leads. On TeleMini v1.0, the power switch leads are soldered directly to the board and can be connected directly to a switch.

For most air-frames, the integrated antennas are more than adequate. However, if you are installing in a carbon-fiber or metal electronics bay which is opaque to RF signals, you may need to use off-board external antennas instead. In this case, you can replace the stock UHF antenna wire with an edge-launched SMA connector, and, on TeleMetrum v1, you can unplug the integrated GPS antenna and select an appropriate off-board GPS antenna with cable terminating in a U.FL connector.

# Chapter 11. Using Altus Metrum Products

## 11.1. Being Legal

In the US, you need an [amateur radio license](#) or other authorization to legally operate the radio transmitters that are part of our products.

## 11.2. 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. An 850mAh battery weighs less than a 9V alkaline battery, and will run a TeleMetrum, TeleMega, EasyMega or EasyTimer for hours. A 110mAh battery weighs less than a triple A battery and is a good choice for use with TeleMini or EasyMini.

By default, we ship TeleMini, TeleMetrum and TeleMega flight computers with a simple wire antenna. If your electronics bay or the air-frame it resides within is made of carbon fiber, which is opaque to RF signals, you may prefer to install an SMA connector so that you can run a coaxial cable to an antenna mounted elsewhere in the rocket. However, note that the GPS antenna is fixed on all current products, so you really want to install the flight computer in a bay made of RF-transparent materials if at all possible.

## 11.3. On the Ground

To receive the data stream from the rocket, you need an antenna and short feed-line connected to one of our [TeleDongle](#) units. If possible, use an SMA to BNC adapter instead of feedline between the antenna feedpoint and TeleDongle, as this will give you the best performance. The TeleDongle in turn plugs directly into the USB port on a notebook computer. Because TeleDongle looks like a simple serial port, your computer does not require special device drivers... just plug it in.

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.

Alternatively, a TeleBT attached with an SMA to BNC adapter at the feed point of a hand-held yagi used in conjunction with an Android device running AltosDroid makes an outstanding ground station.

After the flight, you can use the radio link to extract the more detailed data logged in either TeleMetrum or TeleMini devices, or 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.

If your rocket lands out of sight, you may enjoy having a hand-held GPS receiver, so that you can put in a way-point for the last reported rocket position before touch-down. This makes looking for your rocket a lot like Geo-Caching... just go to the way-point and look around starting from there. AltosDroid on an Android device with GPS receiver works great for this, too!

You may also enjoy having a ham radio “HT” that covers the 70cm band... you can use that with your antenna to direction-find the rocket on the ground the same way you can use a Walston or Beeline tracker. This can be handy if the rocket is hiding in sage brush or a tree, or if the last GPS position doesn't get you close enough because the rocket dropped into a canyon, or the wind is blowing it across a dry lake bed, or something like that... Keith currently uses a Yaesu FT1D, Bdale has a Yaesu VX-7R, which is a nicer radio in most ways but doesn't support APRS.

So, to recap, on the ground the hardware you'll need includes:

1. an antenna and feed-line or adapter
2. a TeleDongle
3. a notebook computer
4. optionally, a hand-held GPS receiver
5. optionally, an HT or receiver covering 435 MHz

The best hand-held commercial directional antennas we've found for radio direction finding rockets are from [Arrow Antennas](#).

The 440-3 and 440-5 are both good choices for finding a TeleMetrum- or TeleMini- equipped rocket when used with a suitable 70cm HT. TeleDongle and an SMA to BNC adapter fit perfectly between the driven element and reflector of Arrow antennas.

## 11.4. 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.

## 11.5. Future Plans

We have designed and prototyped several “companion boards” that can attach to the companion connector on TeleMetrum, TeleMega and EasyMega flight computers to collect more data, provide more pyro channels, and so forth. We do not yet know if or when any of these boards will be produced in enough quantity to sell. If you have specific interests for data collection or control of events in your rockets beyond the capabilities of our existing productions, please let us know!

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 12. AltosUI

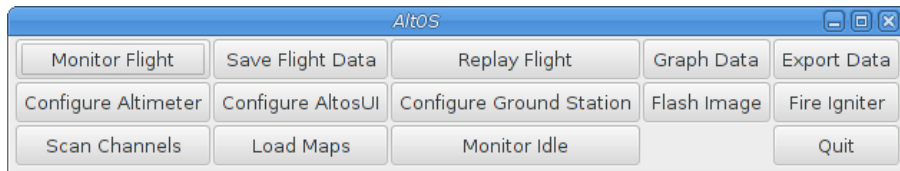


Figure 9. AltosUI Main Window

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.

## 12.1. Monitor Flight

Selecting this item brings up a dialog box listing all of the connected TeleDongle devices. When you choose one of these, AltosUI will create a window to display telemetry data as received by the selected TeleDongle device.



Figure 10. Device Selection Dialog

All telemetry data received are automatically recorded in suitable log files. The name of the files includes the current date and rocket serial and flight numbers.

The radio frequency being monitored by the TeleDongle device is displayed at the top of the window. You can configure the frequency by clicking on the frequency box and selecting the desired frequency. AltosUI remembers the last frequency selected for each TeleDongle and selects that automatically the next time you use that device.

Below the TeleDongle frequency selector, the window contains a few significant pieces of information about the altimeter providing the telemetry data stream:

- The configured call-sign
- The device serial number
- The flight number. Each altimeter remembers how many times it has flown.
- The rocket flight state. Each flight passes through several states including Pad, Boost, Fast, Coast, Drogue, Main and Landed.
- The Received Signal Strength Indicator value. This lets you know how strong a signal TeleDongle is receiving. At the default data rate, 38400 bps, in bench testing, the radio inside

TeleDongle v0.2 operates down to about -106dBm, while the v3 radio works down to about -111dBm. Weaker signals, or an environment with radio noise may cause the data to not be received. The packet link uses error detection and correction techniques which prevent incorrect data from being reported.

- The age of the displayed data, in seconds since the last successfully received telemetry packet. In normal operation this will stay in the low single digits. If the number starts counting up, then you are no longer receiving data over the radio link from the flight computer.

Finally, the largest portion of the window contains a set of tabs, each of which contain some information about the rocket. They're arranged in 'flight order' so that as the flight progresses, the selected tab automatically switches to display data relevant to the current state of the flight. You can select other tabs at any time. The final 'table' tab displays all of the raw telemetry values in one place in a spreadsheet-like format.

### 12.1.1. Launch Pad



Figure 11. Monitor Flight Launch Pad View

The 'Launch Pad' tab shows information used to decide when the rocket is ready for flight. The first elements include red/green indicators, if any of these is red, you'll want to evaluate whether the rocket is ready to launch:

#### Battery Voltage

This indicates whether the Li-Po battery powering the flight computer has sufficient charge to last for the duration of the flight. A value of more than 3.8V is required for a 'GO' status.

#### Apogee Igniter Voltage

This indicates whether the apogee igniter has continuity. If the igniter has a low resistance, then the voltage measured here will be close to the Li-Po battery voltage. A value greater than 3.2V is required for a 'GO' status.

## Main Igniter Voltage

This indicates whether the main igniter has continuity. If the igniter has a low resistance, then the voltage measured here will be close to the Li-Po battery voltage. A value greater than 3.2V is required for a 'GO' status.

## On-board Data Logging

This indicates whether there is space remaining on-board to store flight data for the upcoming flight. If you've downloaded data, but failed to erase flights, there may not be any space left. Most of our flight computers can store multiple flights, depending on the configured maximum flight log size. TeleMini v1.0 stores only a single flight, so it will need to be downloaded and erased after each flight to capture data. This only affects on-board flight logging; the altimeter will still transmit telemetry and fire ejection charges at the proper times even if the flight data storage is full.

## GPS Locked

For a TeleMetrum or TeleMega device, this indicates whether the GPS receiver is currently able to compute position information. GPS requires at least 4 satellites to compute an accurate position.

## GPS Ready

For a TeleMetrum or TeleMega device, this indicates whether GPS has reported at least 10 consecutive positions without losing lock. This ensures that the GPS receiver has reliable reception from the satellites.

The Launchpad tab also shows the computed launch pad position and altitude, averaging many reported positions to improve the accuracy of the fix.

## 12.1.2. Ascent

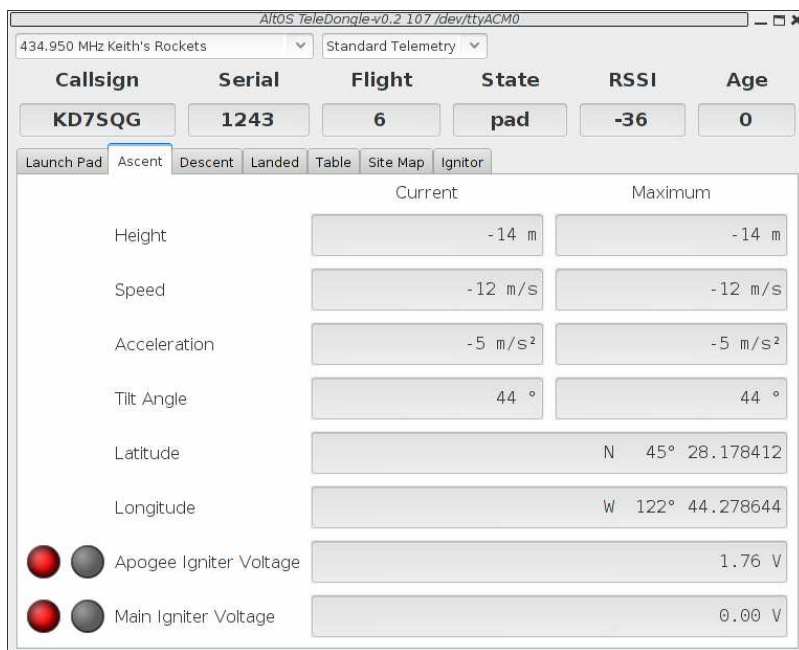


Figure 12. Monitor Flight Ascent View

This tab is shown during Boost, Fast and Coast phases. The information displayed here helps

monitor the rocket as it heads towards apogee.

The height, speed, acceleration and tilt are shown along with the maximum values for each of them. This allows you to quickly answer the most commonly asked questions you'll hear during flight.

The current latitude and longitude reported by the GPS are also shown. Note that under high acceleration, these values may not get updated as the GPS receiver loses position fix. Once the rocket starts coasting, the receiver should start reporting position again.

Finally, the current igniter voltages are reported as in the Launch Pad tab. This can help diagnose deployment failures caused by wiring which comes loose under high acceleration.

### 12.1.3. Descent

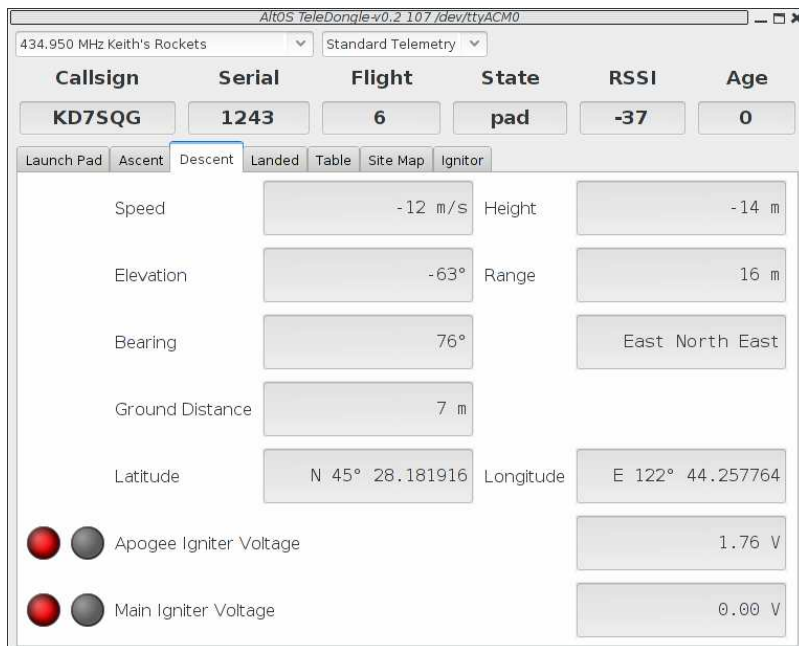


Figure 13. Monitor Flight Descent View

Once the rocket has reached apogee and (we hope) activated the apogee charge, attention switches to tracking the rocket on the way back to the ground, and for dual-deploy flights, waiting for the main charge to fire.

To monitor whether the apogee charge operated correctly, the current descent rate is reported along with the current height. Good descent rates vary based on the choice of recovery components, but generally range from 15-30m/s on drogue and should be below 10m/s when under the main parachute in a dual-deploy flight.

With GPS-equipped flight computers, you can locate the rocket in the sky using the elevation and bearing information to figure out where to look. Elevation is in degrees above the horizon. Bearing is reported in degrees relative to true north. Range can help figure out how big the rocket will appear. Ground Distance shows how far it is to a point directly under the rocket and can help figure out where the rocket is likely to land. Note that all of these values are relative to the pad location. If the elevation is near 90°, the rocket is over the pad, not over you.

Finally, the igniter voltages are reported in this tab as well, both to monitor the main charge as well as to see what the status of the apogee charge is. Note that some commercial e-matches are designed to retain continuity even after being fired, and will continue to show as green or return from red to green after firing.

### 12.1.4. Landed

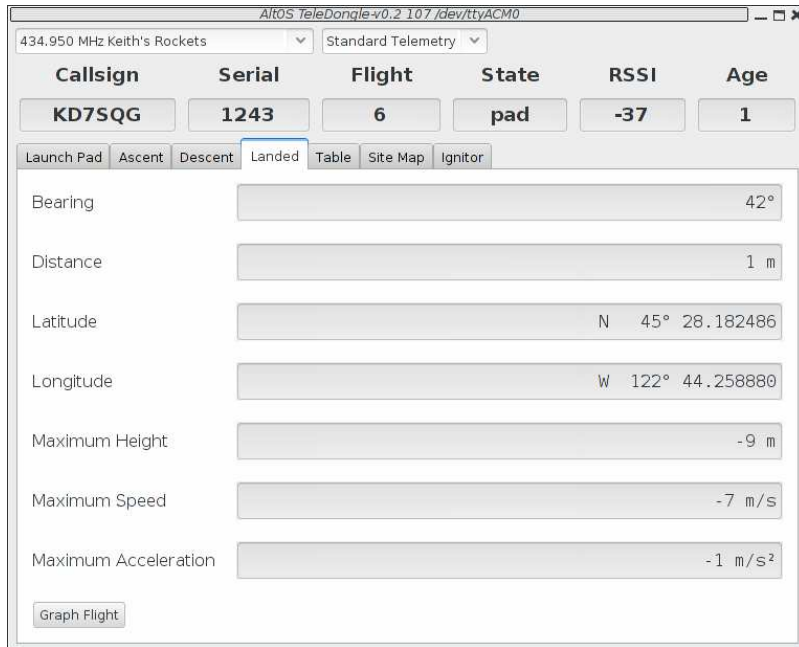


Figure 14. Monitor Flight Landed View

Once the rocket is on the ground, attention switches to recovery. While the radio signal is often lost once the rocket is on the ground, the last reported GPS position is generally within a short distance of the actual landing location.

The last reported GPS position is reported both by latitude and longitude as well as a bearing and distance from the launch pad. The distance should give you a good idea of whether to walk or hitch a ride. Take the reported latitude and longitude and enter them into your hand-held GPS unit and have that compute a track to the landing location.

Our flight computers will continue to transmit RDF tones after landing, allowing you to locate the rocket by following the radio signal if necessary. You may need to get away from the clutter of the flight line, or even get up on a hill (or your neighbor's RV roof) to receive the RDF signal.

The maximum height, speed and acceleration reported during the flight are displayed for your admiring observers. The accuracy of these immediate values depends on the quality of your radio link and how many packets were received. Recovering the on-board data after flight may yield more precise results.

To get more detailed information about the flight, you can click on the 'Graph Flight' button which will bring up a graph window for the current flight.

## 12.1.5. Table

Field	Value	Field	Value	Field	Value
Device	TeleMega	GPS state	ready	Satellites Visible	7
Altitude	52 m	GPS	locked	Satellite id,C/NO	3, 8
Pad altitude	53 m	Satellites	6	Satellite id,C/NO	5, 33
Height	-2 m	Latitude	N 45°28.18956'	Satellite id,C/NO	7, 18
Max height	-2 m	Longitude	W 122°44.28578'	Satellite id,C/NO	8, 22
Acceleration	-1.1 m/s²	GPS altitude	108.0	Satellite id,C/NO	9, 21
Max acceleration	-0.5 m/s²	GPS height	3.0	Satellite id,C/NO	26, 34
Speed	-2.1 m/s	GPS hdop	220.0	Satellite id,C/NO	28, 28
Max Speed	-1.9 m/s	Distance from pad	3 m		
Tilt	4 °	Direction from pad	230°		
Max Tilt	6 °	Elevation from pad	-31°		
Temperature	26.10 °C	Range from pad	4 m		
Battery	4.17 V	Pad latitude	N 45°28.19069'		
Drogue	4.08 V	Pad longitude	W 122°44.28388'		
Main	0.00 V	Pad GPS alt	105 m		
CRC Errors	3	GPS date	2014-01-31		
		GPS time	04:51:21		

Figure 15. Monitor Flight Table View

The table view shows all of the data available from the flight computer. Probably the most useful data on this tab is the detailed GPS information, which includes horizontal dilution of precision information, and information about the signal being received from the satellites.

## 12.1.6. Site Map

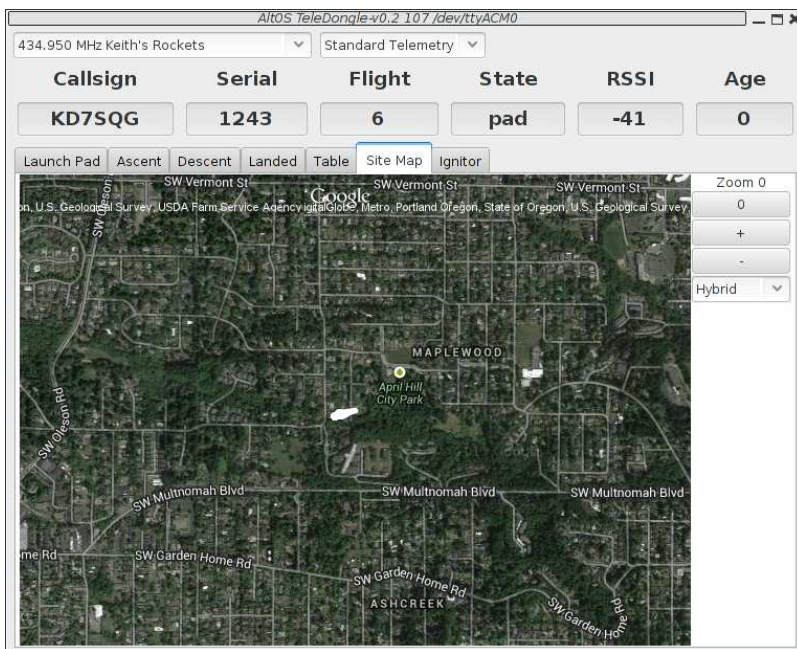


Figure 16. Monitor Flight Site Map View

When the TeleMetrum has a GPS fix, the Site Map tab will map the rocket's position to make it easier for you to locate the rocket, both while it is in the air, and when it has landed. The rocket's state is indicated by color: white for pad, red for boost, pink for fast, yellow for coast, light blue

for drogue, dark blue for main, and black for landed.

The map's default scale is approximately 3m (10ft) per pixel. The map can be dragged using the left mouse button. The map will attempt to keep the rocket roughly centered while data is being received.

You can adjust the style of map and the zoom level with buttons on the right side of the map window. You can draw a line on the map by moving the mouse over the map with a button other than the left one pressed, or by pressing the left button while also holding down the shift key. The length of the line in real-world units will be shown at the start of the line.

Images are fetched automatically via the Google Maps Static API, and cached on disk for reuse. If map images cannot be downloaded, the rocket's path will be traced on a dark gray background instead.

You can pre-load images for your favorite launch sites before you leave home; check out [Load Maps](#).

### 12.1.7. Igniter

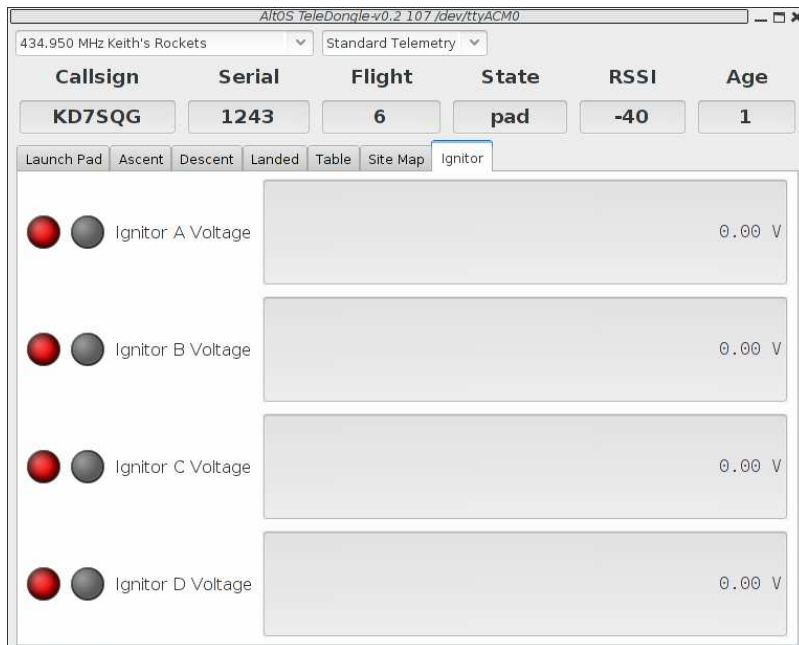


Figure 17. Monitor Flight Additional Igniter View

TeleMega includes four additional programmable pyro channels. The Igniter tab shows whether each of them has continuity. If an ignitor has a low resistance, then the voltage measured here will be close to the pyro battery voltage. A value greater than 3.2V is required for a 'GO' status.

## 12.2. Save Flight Data

The altimeter records flight data to its internal flash memory. Data logged on board is recorded at a much higher rate than the telemetry system can handle, and is not subject to radio drop-outs. As such, it provides a more complete and precise record of the flight. 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. If you select a TeleDongle device, flight data will be downloaded from a flight computer over radio link via the specified TeleDongle. See [Controlling An Altimeter Over The Radio Link](#) for more information.

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.

## 12.3. 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. Check [Monitor Flight](#) to learn how this window operates.

## 12.4. 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.

### 12.4.1. Flight Graph

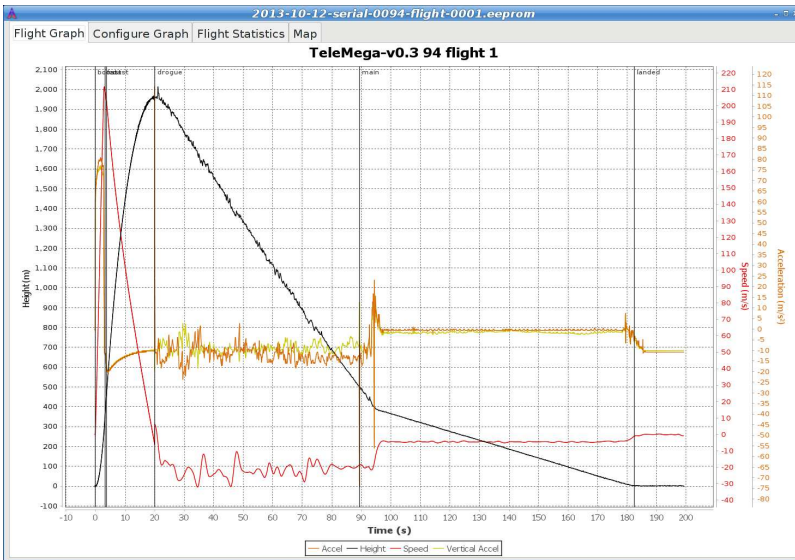


Figure 18. Flight Data Graph

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.

## 12.4.2. Configure Graph

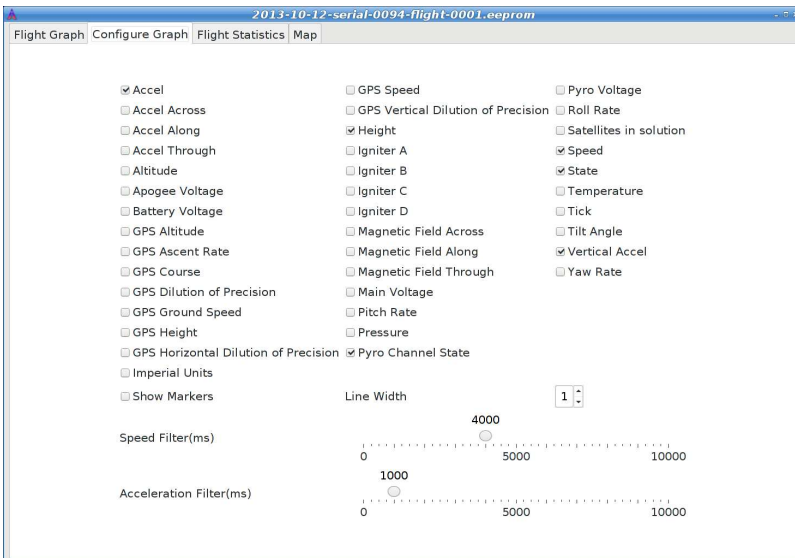


Figure 19. Flight Graph Configuration

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.

### 12.4.3. Flight Statistics

2013-10-12-serial-0094-flight-0001.eeprom			
Flight Graph   Configure Graph   Flight Statistics   Map			
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/s <sup>2</sup>	265 ft/s <sup>2</sup>	8.24 G
Average boost acceleration	65.7 m/s <sup>2</sup>	215 ft/s <sup>2</sup>	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 20. Flight Statistics

Shows overall data computed from the flight.

### 12.4.4. Map

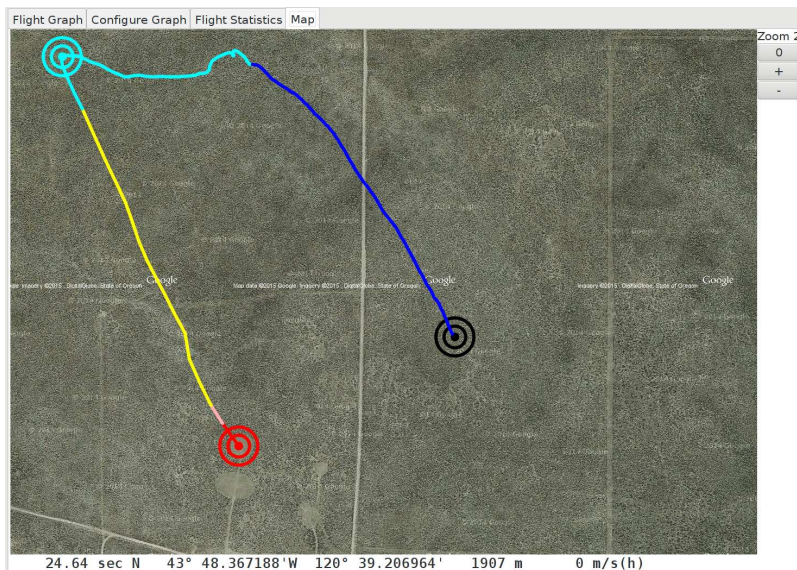


Figure 21. Flight Map

Shows a satellite image of the flight area overlaid with the path of the flight. The flight path will have different colored sections for each state of the flight (just like the Site Map in Monitor Flight

mode): white for pad, red for boost, pink for fast, yellow for coast, light blue for drogue, dark blue for main, and black for landed.

The red concentric circles mark the launch pad, the black concentric circles mark the landing location. Data for the point along the flight path nearest the cursor will be displayed at the bottom of the window. This data includes flight time (so you can correlate data in the graph window), latitude, longitude, height above ground and vertical speed. The selected point will be highlighted with concentric circles in the same color as the flight path at that point.

## 12.5. 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. It has a selector to choose between CSV and KML file formats.

### 12.5.1. 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.

### 12.5.2. Keyhole Markup Language (for Google Earth)

This is the format used by Google Earth to provide an overlay within that application. With this, you can use Google Earth to see the whole flight path in 3D.

## 12.6. Configure Altimeter

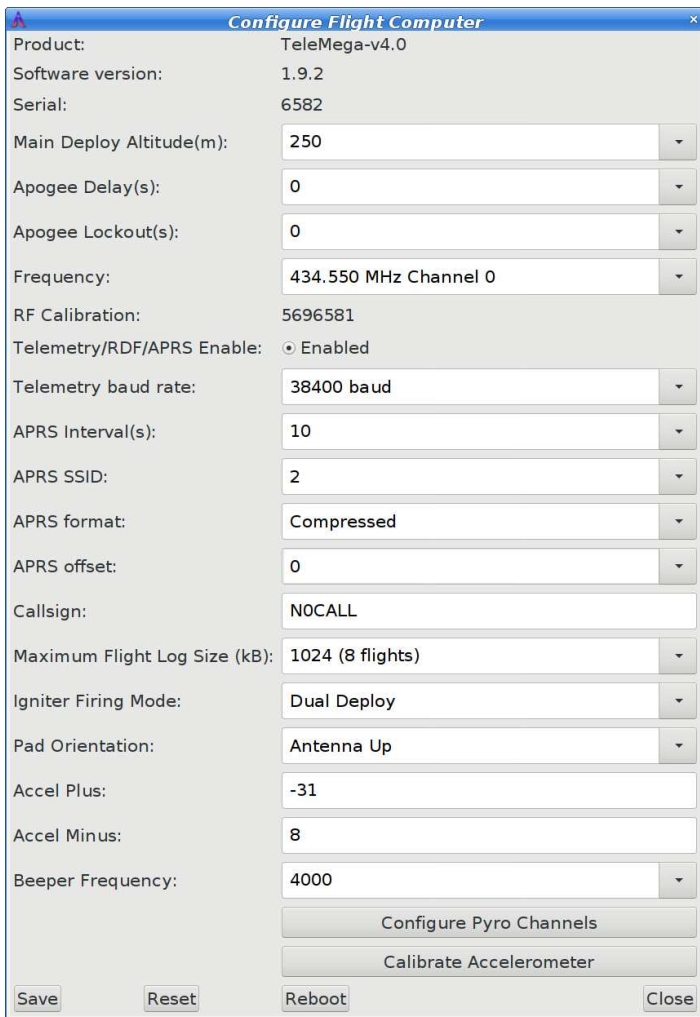


Figure 22. Altimeter Configuration

Select this button and then select either an altimeter or TeleDongle Device from the list provided. Selecting a TeleDongle device will use the radio link to configure a remote 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.

### 12.6.1. 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.

### 12.6.2. 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.

### 12.6.3. 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 occurs 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.



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.

### 12.6.4. Frequency

This configures which of the frequencies to use for both telemetry and packet command mode. Note that if you set this value via packet command mode, the TeleDongle frequency will also be automatically reconfigured to match so that communication will continue afterwards.

### 12.6.5. RF Calibration

The radios in every Altus Metrum device are calibrated at the factory to ensure that they transmit and receive on the specified frequency. If you need to you can adjust the calibration by changing this value. Do not do this without understanding what the value means, read the appendix on calibration and/or the source code for more information. To change a TeleDongle's calibration, you must reprogram the unit completely.

### 12.6.6. Telemetry/RDF/APRS Enable

Enables the radio for transmission during flight. When disabled, the radio will not transmit anything during flight at all.

### 12.6.7. Limit transmit to 10mW

Reduces transmit power to no more than 10mW. This is useful when operating under some UK radio regulations.

### 12.6.8. Telemetry baud rate

This sets the modulation bit rate for data transmission for both telemetry and packet link mode. Lower bit rates will increase range while reducing the amount of data that can be sent and increasing battery consumption. All telemetry is done using a rate 1/2 constraint 4 convolution code, so the actual data transmission rate is 1/2 of the modulation bit rate specified here.

### 12.6.9. APRS Interval

How often to transmit GPS information via APRS (in seconds). When set to zero, APRS transmission is disabled. This option is available on TeleMetrum v2 or newer and TeleMega boards. TeleMetrum v1 boards cannot transmit APRS packets. Note that a single APRS packet takes nearly a full second to transmit, so enabling this option will prevent sending any other telemetry during that time.

### 12.6.10. APRS SSID

Which SSID to report in APRS packets. By default, this is set to the last digit of the serial number, but can be configured to any value from 0 to 9.

### 12.6.11. APRS Format

Whether to send APRS data in Compressed or Uncompressed format. Compressed format is smaller and more precise. Uncompressed format is older, but may work better with your device. The Kenwood TH-D72 only displays altitude information with Uncompressed format, while the Yaesu FT1D only displays altitude with Compressed format. Test before you fly to see which to use.

### 12.6.12. APRS Offset

The delay from the top of the minute before sending the first APRS packet of the minute. Coordinating values for this parameter between multiple devices can allow a single receiver to reliably receive APRS packets from multiple devices. Note that this offset only takes effect while the GPS signal is locked so that the transmitting device knows the current time.

### 12.6.13. Callsign

This sets the call sign included in each telemetry packet. Set this as needed to conform to your local radio regulations.

### 12.6.14. 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.

### 12.6.15. 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.

### 12.6.16. Pad Orientation

Because they include accelerometers, TeleMetrum, TeleMega and EasyMega are sensitive to the orientation of the board. By default, they expect the antenna end to point forward. This parameter allows that default to be changed, permitting the board to be mounted with the antenna pointing aft instead.

#### Antenna Up

In this mode, the antenna (or beeper, for devices without an antenna) of the flight computer must point forward, in line with the expected flight path.

#### Antenna Down

In this mode, the antenna (or beeper, for devices without an antenna) end of the flight computer must point aft, in line with the expected flight path.

### 12.6.17. 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.

## 12.6.18. Logging Trigger Motion

This sets the amount of motion that TeleGPS needs to see before logging the new position. Motions smaller than this are skipped, which saves storage space.

## 12.6.19. Position Reporting Interval

The interval between TeleGPS position reports, both over the air and in the log. Increase this to reduce the frequency of radio transmissions and the length of time available in the log.

## 12.6.20. Calibrate Accelerometer

This opens a separate window to recalibrate the accelerometers. Follow the instructions, orienting the flight computer with the antenna end, or end opposite the screw terminals, in the case of EasyMega, first up and then down.

When the calibration is complete, return to the Configure Altimeter window and save the new calibration values.

## 12.6.21. Configure Pyro Channels

The screenshot shows a window titled "Configure Pyro Channels" with a close button in the top right corner. The window is organized into a grid with four columns representing Pyro Channel A, B, C, and D. The rows represent various conditions for triggering a pyro channel. Each cell in the grid contains a checkbox, a text input field, and a dropdown menu. The "Pyro Firing Time(s)" is set to 0.05 at the bottom. A "Close" button is located at the bottom right of the window.

	Pyro Channel A	Pyro Channel B	Pyro Channel C	Pyro Channel D
Acceleration less than (m/s <sup>2</sup> )	<input checked="" type="checkbox"/> 1.1	<input checked="" type="checkbox"/> 0.0	<input type="checkbox"/> 0.0	<input type="checkbox"/> 0.0
Acceleration greater than (m/s <sup>2</sup> )	<input checked="" type="checkbox"/> 2.2	<input checked="" type="checkbox"/> 0.0	<input type="checkbox"/> 0.0	<input type="checkbox"/> 0.0
Speed less than (m/s)	<input checked="" type="checkbox"/> 3.3	<input checked="" type="checkbox"/> 0.0	<input type="checkbox"/> 0.0	<input type="checkbox"/> 0.0
Speed greater than (m/s)	<input checked="" type="checkbox"/> 4.4	<input checked="" type="checkbox"/> 0.0	<input type="checkbox"/> 0.0	<input type="checkbox"/> 0.0
Height less than (m)	<input checked="" type="checkbox"/> 5	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
Height greater than (m)	<input checked="" type="checkbox"/> 6	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
Angle from vertical less than (degrees)	<input checked="" type="checkbox"/> 7	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
Angle from vertical greater than (degrees)	<input checked="" type="checkbox"/> 8	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
Time since boost less than (s)	<input checked="" type="checkbox"/> 9.90	<input type="checkbox"/> 0.00	<input type="checkbox"/> 0.00	<input type="checkbox"/> 0.00
Time since boost greater than (s)	<input checked="" type="checkbox"/> 10.10	<input type="checkbox"/> 0.00	<input type="checkbox"/> 0.00	<input type="checkbox"/> 0.00
Ascending	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Descending	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
After motor number	<input checked="" type="checkbox"/> 11	<input type="checkbox"/> 0	<input type="checkbox"/> 0	<input type="checkbox"/> 0
Delay after other conditions (s)	<input checked="" type="checkbox"/> 12.12	<input checked="" type="checkbox"/> 1.00	<input type="checkbox"/> 0.00	<input type="checkbox"/> 0.00
Flight state before	<input checked="" type="checkbox"/> Landed	<input type="checkbox"/> Boost	<input type="checkbox"/> Boost	<input type="checkbox"/> Boost
Flight state after	<input checked="" type="checkbox"/> Drogue	<input checked="" type="checkbox"/> Main	<input type="checkbox"/> Boost	<input type="checkbox"/> Boost
Pyro Firing Time(s):	0.05			

Figure 23. Additional Pyro Channel Configuration

This opens a separate window to configure the additional pyro channels available on TeleMega, EasyMega and EasyTimer. One column is presented for each channel. Each row represents a single parameter, if enabled the parameter must meet the specified test for the pyro channel to be fired.

Select conditions and set the related value; the pyro channel will be activated when **all** of the conditions are met. Each pyro channel has a separate set of configuration values, so you can use different values for the same condition with different channels.

At the bottom of the window, the 'Pyro Firing Time' configuration sets the length of time (in seconds) which each of these pyro channels will fire for.

Once you have selected the appropriate configuration for all of the necessary pyro channels, you can save the pyro configuration along with the rest of the flight computer configuration by pressing the 'Save' button in the main Configure Flight Computer window.

### Vertical Acceleration

Select a value, and then choose whether acceleration away from the ground should be above or below that value. Acceleration is positive upwards, so accelerating towards the ground would produce negative numbers. Acceleration during descent is noisy and inaccurate, so be careful when using it during these phases of the flight.

### Ascent rate

Select a value, and then choose whether ascent rate should be above or below that value. Ascent rate is positive upwards, so moving towards the ground would produce negative numbers. Ascent rate during descent is a bit noisy and so be careful when using it during these phases of the flight.

### Height above pad

Select a value, and then choose whether the height above the launch pad should be above or below that value. Note that because EasyTimer has only a low-range accelerometer and no barometer, this value will not be very reliable on that device.

### Orientation

TeleMega, EasyMega and EasyTimer contain a 3-axis gyroscope and accelerometer which is used to compute the orientation of the rocket. A record of orientations over the last 0.64 seconds is kept and the largest value within this period is compared with the specified value. Note that the tilt angle is not the change in angle from the launch pad, but rather absolute relative to gravity—the 3-axis accelerometer is used to compute the angle of the rocket on the launch pad and initialize the system.



Because this value is computed by integrating rate gyros, it gets progressively less accurate as the flight goes on. It should have an accumulated error of less than 0.2°/second (after 10 seconds of flight, the error should be less than 2°).

The usual use of the orientation configuration is to ensure that the rocket is traveling mostly upwards when deciding whether to ignite air starts or additional stages. For that, choose a reasonable maximum angle (like 20°) and set the motor igniter to require an angle of less than that value.

### Flight Time

Time since launch. Select a value and choose whether to activate the pyro channel before or after that amount of time.



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.



Firmware versions older than 1.9.8 cannot use times longer than 327.67

seconds. Update firmware if you need a longer time.

## Ascending

A deprecated configuration value which was the same as setting Ascent rate  $> 0$ . Existing configurations using this will be cleared and must be reconfigured by the user.

## Descending

A deprecated configuration value which was the same as setting Ascent rate  $< 0$ . Existing configurations using this will be cleared and must be reconfigured by the user.

## After Motor

The flight software counts each time the rocket starts accelerating and then decelerating (presumably due to a motor or motors burning). Use this value for multi-staged or multi-airstart launches. As of version 1.8.6 firmware, this checks to make sure at least this many motors have burned. Before version 1.8.6, this checked to make sure that exactly this many motors had burned.

## Delay

Once the other parameters all become true, a timer is started for the specified amount of time. While the timer is running, the other parameters are checked repeatedly and if any of them become false, then the pyro channel is disabled and will not fire. If the timer expires and all of the other parameters have remained true for the entire time, then the pyro channel is fired.



Firmware versions older than 1.9.8 cannot use delays longer than 327.67 seconds. Update firmware if you need a longer delay.

## Flight State

The flight software tracks the flight through a sequence of states:

- Boost. The motor has lit and the rocket is accelerating upwards. Ascent rate will be greater than zero. Vertical acceleration will be greater than zero.
- Fast. The motor has burned out and the rocket is decelerating, but it is going faster than 200m/s. Ascent rate will be greater than zero. Vertical acceleration will be less than zero.
- Coast. The rocket is still moving upwards and decelerating, but the Ascent rate is less than 200m/s. Ascent rate will be greater than zero. Vertical acceleration will be less than zero.
- Drogue. The rocket has reached apogee and is heading back down, but is above the configured Main altitude. Ascent rate will be less than zero during this state. Vertical acceleration will be negative until the rocket reaches a terminal descent rate, at which point Vertical acceleration will be zero. Both Ascent rate and Vertical acceleration are very noisy in this state, so be careful when trying to use them to control pyro channels. This state selection is not available on EasyTimer.
- Main. The rocket is still descending, and is below the Main altitude. Ascent rate will be less than zero during this state. Vertical acceleration may be briefly less than zero as the rocket slows from drogue descent to main descent, but it will settle down to a zero value once the rocket has reached the terminal velocity under the main chute. Ascent rate and

Vertical acceleration should be much less noisy once the main chute has deployed. This state selection is not available on EasyTimer.

- Landed. The rocket is no longer moving.

You can select a state to limit when the pyro channel may activate; note that the check is based on when the rocket transitions **into** the state, and so checking for “greater than Boost” means that the rocket is currently in boost or some later state.

When a motor burns out, the rocket enters either Fast or Coast state (depending on how fast it is moving). If the computer detects upwards acceleration again, it will move back to Boost state.

## 12.7. Configure AltosUI



Figure 24. Configure AltosUI Dialog

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

### 12.7.1. Voice Settings

AltosUI provides voice announcements during flight so that you can keep your eyes on the sky and still get information about the current flight status. However, sometimes you don't want to hear them.

#### Enable

Turns all voice announcements on and off

#### Test Voice

Plays a short message allowing you to verify that the audio system is working and the volume settings are reasonable

### 12.7.2. Log Directory

AltosUI logs all telemetry data and saves all flash data to this directory. This directory is also

used as the starting 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.

### 12.7.3. Callsign

This value is transmitted in each command packet sent from TeleDongle and received from an altimeter. It is not used in telemetry mode, as the callsign configured in the altimeter board is included in all telemetry packets. Configure this with the AltosUI operators call sign as needed to comply with your local radio regulations.

Note that to successfully command a flight computer over the radio (to configure the altimeter, monitor idle, or fire pyro charges), the callsign configured here must exactly match the callsign configured in the flight computer. This matching is case sensitive.

### 12.7.4. 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.

### 12.7.5. 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.

### 12.7.6. Font size

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

### 12.7.7. 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.

### 12.7.8. Menu position

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

### 12.7.9. Map Cache Size

Sets the number of map 'tiles' kept in memory while the application is running. More tiles consume more memory, but will make panning around the map faster.

## 12.7.10. Manage Frequencies

This brings up a dialog where you can configure the set of frequencies shown in the various frequency menus. You can add as many as you like, or even reconfigure the default set. Changing this list does not affect the frequency settings of any devices, it only changes the set of frequencies shown in the menus.

## 12.8. Configure Groundstation

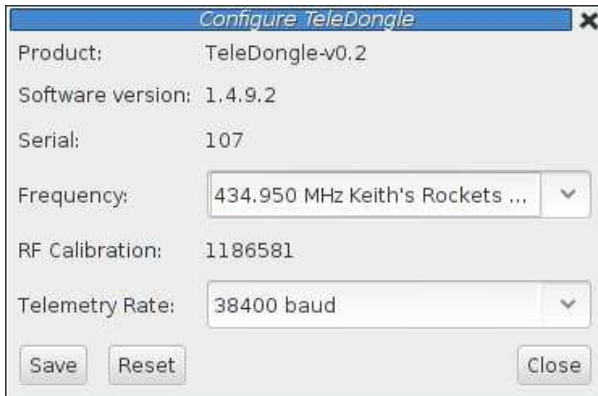


Figure 25. Configure Groundstation Dialog

Select this button and then select a TeleDongle or TeleBT Device from the list provided.

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.

Note that TeleDongle and TeleBT don't save any configuration data, the settings here are recorded on the local machine in the Java preferences database. Moving the device to another machine, or using a different user account on the same machine will cause settings made here to have no effect.

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

### Save

This writes any changes to the local Java preferences file. 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.

### Close

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

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

### 12.8.1. Frequency

This configures the frequency to use for both telemetry and packet command mode. Set this

before starting any operation involving packet command mode so that it will use the right frequency. Telemetry monitoring mode also provides a menu to change the frequency, and that menu also sets the same Java preference value used here.

### 12.8.2. RF Calibration

The radios in every Altus Metrum device are calibrated at the factory to ensure that they transmit and receive on the specified frequency. To change a TeleDongle or TeleBT's calibration, you must reprogram the unit completely, so this entry simply shows the current value and doesn't allow any changes.

### 12.8.3. Telemetry Rate

This lets you match the telemetry and packet link rate from the transmitter. If they don't match, the device won't receive any data.

## 12.9. Flash Image

This reprograms Altus Metrum devices with new firmware. TeleMetrum v1.x, TeleDongle v0.2, TeleMini v1.0 and TeleBT v1.0 are all reprogrammed by using another similar unit as a programming dongle (pair programming). TeleMega, EasyMega, TeleMetrum v2 or newer, EasyMini, EasyTimer, TeleBT v3 or newer and TeleDongle v3 or newer are all programmed directly over USB (self programming). Please read the directions for flashing devices in [Updating Device Firmware](#).

## 12.10. Fire Igniter

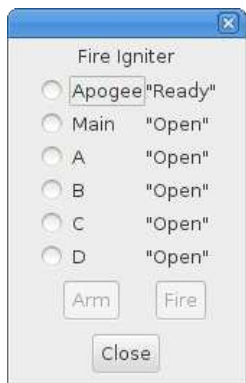


Figure 26. Fire Igniter Window

This activates the igniter circuits in the flight computer to help test recovery systems deployment. Because this command can operate over the Packet Command Link, you can prepare the rocket as for flight and then test the recovery system without needing to snake wires inside the air-frame.

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.

## 12.11. Scan Channels



Figure 27. Scan Channels Window

This listens for telemetry packets on all of the configured frequencies, displaying information about each device it receives a packet from. You can select which of the baud rates and telemetry formats should be tried; by default, it only listens at 38400 baud with the standard telemetry format used in v1.0 and later firmware.

## 12.12. Load Maps

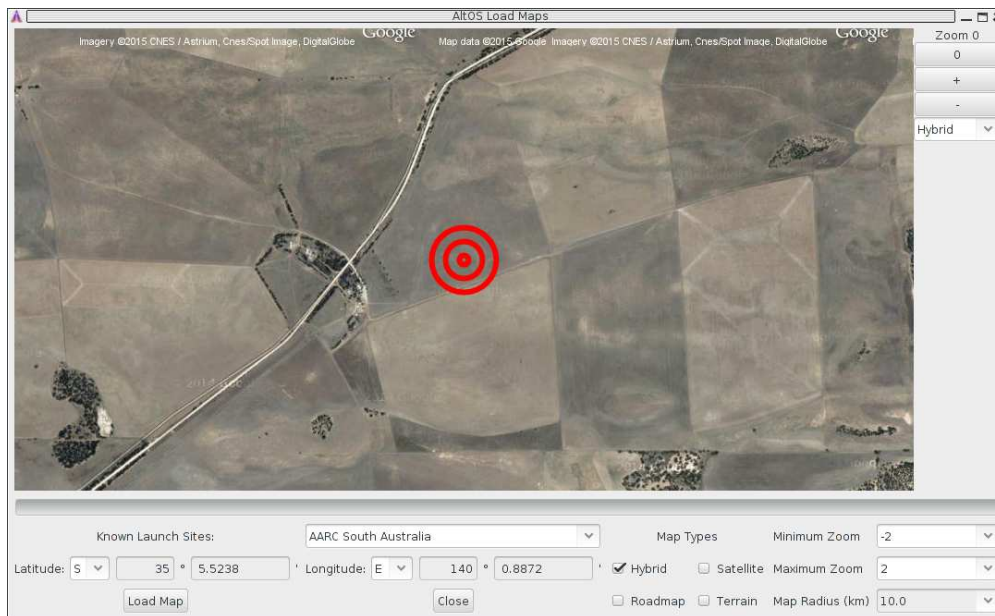


Figure 28. Load Maps Window

Before heading out to a new launch site, you can use this to load satellite images in case you don't have internet connectivity at the site. Try not to wait until the last minute, though, particularly if you're heading to a major launch. If too many people are all trying to download

map data at once, Google may limit access until the next day.

There's a drop-down menu of launch sites we know about; if your favorites aren't there, please let us know the lat/lon and name of the site. The contents of this list are actually downloaded from our server at run-time, so as new sites are sent in, they'll get automatically added to this list. If the launch site isn't in the list, you can manually enter the lat/lon values

There are four different kinds of maps you can view; you can select which to download by selecting as many as you like from the available types:

### **Hybrid**

A combination of satellite imagery and road data. This is the default view.

### **Satellite**

Just the satellite imagery without any annotation.

### **Roadmap**

Roads, political boundaries and a few geographic features.

### **Terrain**

Contour intervals and shading that show hills and valleys.

You can specify the range of zoom levels to download; smaller numbers show more area with less resolution. The default level, 0, shows about 3m/pixel. One zoom level change doubles or halves that number. Larger zoom levels show more detail, smaller zoom levels less.

The Map Radius value sets how large an area around the center point to download. Select a value large enough to cover any plausible flight from that site. Be aware that loading a large area with a high maximum zoom level can attempt to download a lot of data. Loading hybrid maps with a 10km radius at a minimum zoom of -2 and a maximum zoom of 2 consumes about 120MB of space. Terrain and road maps consume about 1/10 as much space as satellite or hybrid maps.

Clicking the 'Load Map' button will fetch images from Google Maps; note that Google limits how many images you can fetch at once, so if you load more than one launch site, you may get some gray areas in the map which indicate that Google is tired of sending data to you. Try again later.

## **12.13. Monitor Idle**

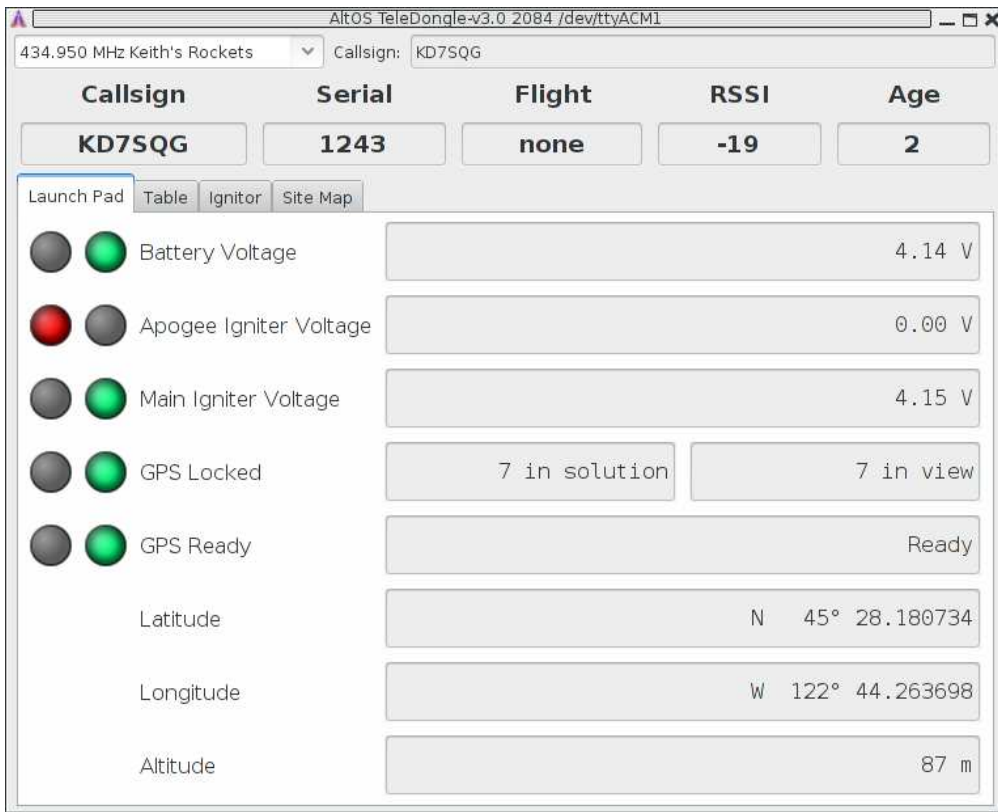


Figure 29. Monitor Idle Window

This brings up a dialog similar to the Monitor Flight UI, except it works with the altimeter in “idle” mode by sending query commands to discover the current state rather than listening for telemetry packets. Because this uses command mode, it needs to have the TeleDongle and flight computer callsigns match exactly. If you can receive telemetry, but cannot manage to run Monitor Idle, then it’s very likely that your callsigns are different in some way.

You can change the frequency and callsign used to communicate with the flight computer; they must both match the configuration in the flight computer exactly.

# Chapter 13. AltosDroid

AltosDroid provides the same flight monitoring capabilities as AltosUI, but runs on Android devices. AltosDroid is designed to connect to a TeleBT receiver over Bluetooth™ and (on Android devices supporting USB On-the-go) TeleDongle and TeleBT devices over USB. AltosDroid monitors telemetry data, logging it to internal storage in the Android device, and presents that data in a UI similar to the 'Monitor Flight' window in AltosUI.

This manual will explain how to configure AltosDroid, connect to TeleBT or TeleDongle, operate the flight monitoring interface and describe what the displayed data means.

## 13.1. Installing AltosDroid

AltosDroid is available from the Google Play store. To install it on your Android device, open the Google Play Store application and search for “altosdroid”. Make sure you don't have a space between “altos” and “droid” or you probably won't find what you want. That should bring you to the right page from which you can download and install the application.

## 13.2. Charging TeleBT Battery

Before using TeleBT with AltosDroid, make sure the internal TeleBT battery is charged. To do this, attach a micro USB cable from a computer or other USB power source to TeleBT. A dual LED on the circuit board should illuminate, showing red while the battery is charging, green when charging is completed, and both red and green on at the same time if there is a battery fault.

## 13.3. Connecting to TeleBT over Bluetooth™

Note that when turning TeleBT on, you may see a brief LED flash, but there will be no "activity" indicated until you pair with the device from AltosDroid.

Press the Android 'Menu' button or soft-key to see the configuration options available. Select the 'Connect a device' option and then the 'Scan for devices' entry at the bottom to look for your TeleBT device. Select your device, and when it asks for the code, enter '1234'.

Subsequent connections will not require you to enter that code, and your 'paired' device will appear in the list without scanning.

## 13.4. Connecting to TeleDongle or TeleBT over USB

Get a special USB On-the-go adapter cable. These cables have a USB micro-B male connector on one end and a standard A female connector on the other end. Plug in your TeleDongle or TeleBT device to the adapter cable and the adapter cable into your phone and AltosDroid should automatically start up. If it doesn't, the most likely reason is that your Android device doesn't support USB On-the-go.

## 13.5. AltosDroid Menu

The main AltosDroid menu has a selection of operation and configuration options.

### Connect a device

Offers a menu of available TeleBT devices, and an option to scan for additional devices.

### Disconnect device

Disconnects the current TeleBT or TeleDongle device. You can reconnect TeleBT over bluetooth by using Connect a Device. You can reconnect a USB device by unplugging it and then plugging it back in.

### Select radio frequency

This selects which frequency to listen on by bringing up a menu of pre-set radio frequencies. Pick the one which matches your altimeter.

### Select Tracker

Switches the information displays to show data for a different transmitting device. The map will always show all of the devices in view. Trackers are shown and selected by serial number, so make sure you note the serial number of devices in each airframe.

### Delete Track

Deletes all information about a transmitting device.

### Setup

Offers additional configuration operations. See [Setup](#).

### Idle Mode

Communicate with an altimeter in Idle mode. See [Idle Mode](#)

### Quit

Shuts down AltosDroid.

## 13.6. Setup

### Telemetry Rate

Altus Metrum transmitters can be configured to operate at lower data rates to improve transmission range. If you have configured your device to do this, this menu item allows you to change the receiver to match.

### Units

Selects which units used to report values.

### Map Type

Displays a menu of map types and lets you select one. Hybrid maps include satellite images with a roadmap overlaid. Satellite maps dispense with the roadmap overlay. Roadmap shows just the roads. Terrain includes roads along with shadows indicating changes in elevation, and

other geographical features.

### **Map Source**

Select between online and offline maps. Online maps will show a 'move to current position' icon in the upper right corner, while offline maps will have copyright information all over the map. Otherwise, they're pretty similar.

### **Preload Maps**

Brings up a dialog allowing you to download offline map tiles so that you can have maps available even if you have no network connectivity at the launch site.

### **Manage Frequencies**

This presents a dialog containing the current list of frequencies that will be show in the 'Select radio frequency' menu. You can change the label for existing frequencies, delete or add new frequencies.

## **13.7. Idle Mode**

### **Callsign**

Lets you configure the callsign used by AltosDroid to communicate with the flight computer. The callsign on the two devices must match or the communication will fail. This provides a modest amount of protection against accidentally controlling another persons flight computer.

### **Monitor**

This provides similar information to monitoring the telemetry stream from a flight computer in flight mode. The title of the application will change to include (idle) so you can tell this mode is active.

### **Reboot**

Remotely reboots the flight computer. This is useful when the flight computers turned on while the airframe is horizontal and you want to restart the flight computers in pad mode after raising the airframe to vertical.

### **Fire Igniters**

Remotely control igniters for ground testing recovery systems.

## **13.8. AltosDroid Flight Monitoring**

AltosDroid is designed to mimic the AltosUI flight monitoring display, providing separate tabs for each stage of your rocket flight along with a tab containing a map of the local area with icons marking the current location of the altimeter and the Android device.

## **13.9. Pad**

The 'Pad' tab shows information used to decide when the rocket is ready for flight. The first elements include red/green indicators, if any of these is red, you'll want to evaluate whether the

rocket is ready to launch.

When the pad tab is selected, the voice responses will include status changes to the igniters and GPS reception, letting you know if the rocket is still ready for launch.

### **Battery**

This indicates whether the Li-Po battery powering the transmitter has sufficient charge to last for the duration of the flight. A value of more than 3.8V is required for a 'GO' status.

### **Receiver Battery**

This indicates whether the Li-Po battery powering the TeleBT has sufficient charge to last for the duration of the flight. A value of more than 3.8V is required for a 'GO' status.

### **Data Logging**

This indicates whether there is space remaining on-board to store flight data for the upcoming flight. If you've downloaded data, but failed to erase flights, there may not be any space left. TeleMetrum and TeleMega can store multiple flights, depending on the configured maximum flight log size. TeleGPS logs data continuously. TeleMini v1.0 stores only a single flight, so it will need to be downloaded and erased after each flight to capture data. This only affects on-board flight logging; the altimeter will still transmit telemetry and fire ejection charges at the proper times.

### **GPS Locked**

For a TeleMetrum or TeleMega device, this indicates whether the GPS receiver is currently able to compute position information. GPS requires at least 4 satellites to compute an accurate position.

### **GPS Ready**

For a TeleMetrum or TeleMega device, this indicates whether GPS has reported at least 10 consecutive positions without losing lock. This ensures that the GPS receiver has reliable reception from the satellites.

### **Apogee Igniter**

This indicates whether the apogee igniter has continuity. If the igniter has a low resistance, then the voltage measured here will be close to the Li-Po battery voltage. A value greater than 3.2V is required for a 'GO' status.

### **Main Igniter**

This indicates whether the main igniter has continuity. If the igniter has a low resistance, then the voltage measured here will be close to the Li-Po battery voltage. A value greater than 3.2V is required for a 'GO' status.

### **Igniter A-D**

This indicates whether the indicated additional pyro channel igniter has continuity. If the igniter has a low resistance, then the voltage measured here will be close to the Li-Po battery voltage. A value greater than 3.2V is required for a 'GO' status.

The Pad tab also shows the location of the Android device.

## 13.10. Flight

The 'Flight' tab shows information used to evaluate and spot a rocket while in flight. It displays speed and height data to monitor the health of the rocket, along with elevation, range and bearing to help locate the rocket in the sky.

While the Flight tab is displayed, the voice announcements will include current speed, height, elevation and bearing information.

### Speed

Shows current vertical speed. During descent, the speed values are averaged over a fairly long time to try and make them steadier.

### Height

Shows the current height above the launch pad.

### Max Speed

Shows the maximum vertical speed seen during the flight.

### Max Height

Shows the maximum height above launch pad.

### Elevation

This is the angle above the horizon from the android devices current position.

### Range

The total distance from the android device to the rocket, including both ground distance and difference in altitude. Use this to gauge how large the rocket is likely to appear in the sky.

### Bearing

This is the azimuth from true north for the rocket from the android device. Use this in combination with the Elevation value to help locate the rocket in the sky, or at least to help point the antenna in the general direction. This is provided in both degrees and a compass point (like West South West). You'll want to know which direction is true north before launching your rocket.

### Ground Distance

This shows the distance across the ground to the lat/lon where the rocket is located. Use this to estimate what is currently under the rocket.

### Latitude/Longitude

Displays the last known location of the rocket.

### Apogee Igniter

This indicates whether the apogee igniter has continuity. If the igniter has a low resistance, then the voltage measured here will be close to the Li-Po battery voltage. A value greater than 3.2V is required for a 'GO' status.

## Main Igniter

This indicates whether the main igniter has continuity. If the igniter has a low resistance, then the voltage measured here will be close to the Li-Po battery voltage. A value greater than 3.2V is required for a 'GO' status.

## 13.11. Recover

The 'Recover' tab shows information used while recovering the rocket on the ground after flight.

While the Recover tab is displayed, the voice announcements will include distance along with either bearing or direction, depending on whether you are moving.

### Bearing

This is the azimuth from true north for the rocket from the android device. Use this in combination with the Elevation value to help locate the rocket in the sky, or at least to help point the antenna in the general direction. This is provided in both degrees and a compass point (like West South West). You'll want to know which direction is true north before launching your rocket.

### Direction

When you are in motion, this provides the angle from your current direction of motion towards the rocket.

### Distance

Distance over the ground to the rocket.

### Tar Lat/Tar Lon

Displays the last known location of the rocket.

### My Lat/My Lon

Displays the location of the Android device.

### Max Height

Shows the maximum height above launch pad seen during the flight.

### Max Speed

Shows the maximum vertical speed seen during the flight.

### Max Accel

Shows the maximum vertical acceleration seen during the flight.

## 13.12. Map

The 'Map' tab shows a map of the area around the rocket being tracked along with information needed to recover it.

On the map itself, icons showing the location of the android device along with the last known location of each tracker. A blue line is drawn from the android device location to the currently

selected tracker.

Below the map, the distance and either bearing or direction along with the lat/lon of the target and the android device are shown

The Map tab provides the same voice announcements as the Recover tab.

## 13.13. Downloading Flight Logs

AltosDroid always saves every bit of telemetry data it receives. To download that to a computer for use with AltosUI, remove the SD card from your Android device, or connect your device to your computer's USB port and browse the files on that device. You will find '.telem' files in the TeleMetrum directory that will work with AltosUI directly.

# 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. For TeleMetrum, TeleMega, EasyMega and EasyTimer, which have accelerometers, the mode is controlled by the orientation of the rocket (well, actually the board, of course...) at the time power is switched on. If the rocket is “nose up”, then the flight computer assumes it’s on a rail or rod being prepared for launch, so the firmware chooses flight mode. However, if the rocket is more or less horizontal, the firmware instead enters idle mode. 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. This is also how EasyMotor works, since even though it has an accelerometer, it detects the motor burn it is meant to record by detecting a rise in chamber pressure, and board orientation doesn’t matter. TeleMini selects “idle” mode if it receives a command packet within the first five seconds of operation.

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. The altimeters also listen for the radio link when in idle mode for requests sent via TeleDongle. Commands can be issued in idle mode over either USB or the radio link equivalently. TeleMini only has the radio link. 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.

One “neat trick” of particular value when TeleMetrum, TeleMega or EasyMega are used with very large air-frames, is that you can power the board up while the rocket is horizontal, such that it comes up in idle mode. Then you can raise the air-frame to launch position, and issue a 'reset' command via TeleDongle over the radio link to cause the altimeter to reboot and come up in flight mode. This is much safer than standing on the top step of a rickety step-ladder or hanging off the side of a launch tower with a screw-driver trying to turn on your avionics before installing igniters!

TeleMini is configured solely via the radio link. Of course, that means you need to know the TeleMini radio configuration values or you won't be able to communicate with it. For situations when you don't have the radio configuration values, TeleMini v1.0 offers an 'emergency recovery' mode. In this mode, TeleMini v1.0 is configured as follows:

- Sets the radio frequency to 434.550MHz
- Sets the radio calibration back to the factory value.
- Sets the callsign to NOCALL
- Does not go to 'pad' mode after five seconds.

To get into 'emergency recovery' mode, first find the row of four small holes opposite the switch wiring. Using a short piece of small gauge wire, connect the outer two holes together, then power TeleMini up. Once the red LED is lit, disconnect the wire and the board should signal that it's in 'idle' mode after the initial five second startup period.

## A.2. GPS

TeleMetrum and TeleMega include a complete GPS receiver. A complete explanation of how GPS works is beyond the scope of this manual, but the bottom line is that the GPS receiver needs to lock onto at least four satellites to obtain a solid 3 dimensional position fix and know what time it is.

The flight computers provide backup power to the GPS chip any time a battery is connected. This allows the receiver to “warm start” on the launch rail much faster than if every power-on were a GPS “cold start”. In typical operations, powering up on the flight line in idle mode while performing final air-frame preparation will be sufficient to allow the GPS receiver to cold start and acquire lock. Then the board can be powered down during RSO review and installation on a launch rod or rail. When the board is turned back on, the GPS system should lock very quickly, typically long before igniter installation and return to the flight line are complete.

## A.3. Controlling An Altimeter Over The Radio Link

One of the unique features of the Altus Metrum system is the ability to create a two way command link between TeleDongle and an altimeter using the digital radio transceivers built into each device. This allows you to interact with the altimeter from afar, as if it were directly connected to the computer.

Any operation which can be performed with a flight computer can either be done with the

device directly connected to the computer via the USB cable, or through the radio link. TeleMini doesn't provide a USB connector and so it is always communicated with over radio. Select the appropriate TeleDongle device when the list of devices is presented and AltosUI will interact with an altimeter over the radio link.

One oddity in the current interface is how AltosUI selects the frequency for radio communications. Instead of providing an interface to specifically configure the frequency, it uses whatever frequency was most recently selected for the target TeleDongle device in Monitor Flight mode. If you haven't ever used that mode with the TeleDongle in question, select the Monitor Flight button from the top level UI, and pick the appropriate TeleDongle device. Once the flight monitoring window is open, select the desired frequency and then close it down again. All radio communications will now use that frequency.

- Save Flight Data—Recover flight data from the rocket without opening it up.
- Configure altimeter apogee delays, main deploy heights and additional pyro event conditions to respond to changing launch conditions. You can also 'reboot' the altimeter. Use this to remotely enable the flight computer by turning TeleMetrum or TeleMega on in "idle" mode, then once the air-frame is oriented for launch, you can reboot the altimeter and have it restart in pad mode without having to climb the scary ladder.
- Fire Igniters—Test your deployment charges without snaking wires out through holes in the air-frame. Simply assemble the rocket as if for flight with the apogee and main charges loaded, then remotely command the altimeter to fire the igniters.

Operation over the radio link for configuring an altimeter, ground testing igniters, and so forth uses the same RF frequencies as flight telemetry. To configure the desired TeleDongle frequency, select the monitor flight tab, then use the frequency selector and close the window before performing other desired radio operations.

The flight computers only enable radio commanding in 'idle' mode. TeleMetrum and TeleMega use the accelerometer to detect which orientation they start up in, so make sure you have the flight computer lying horizontally when you turn it on. Otherwise, it will start in 'pad' mode ready for flight, and will not be listening for command packets from TeleDongle.

TeleMini listens for a command packet for five seconds after first being turned on, if it doesn't hear anything, it enters 'pad' mode, ready for flight and will no longer listen for command packets. The easiest way to connect to TeleMini is to initiate the command and select the TeleDongle device. At this point, the TeleDongle will be attempting to communicate with the TeleMini. Now turn TeleMini on, and it should immediately start communicating with the TeleDongle and the desired operation can be performed.

You can monitor the operation of the radio link by watching the lights on the devices. The red LED will flash each time a packet is transmitted, while the green LED will light up on TeleDongle when it is waiting to receive a packet from the altimeter.

## A.4. Ground Testing

An important aspect of preparing a rocket using electronic deployment for flight is ground testing the recovery system. Thanks to the bi-directional radio link central to the Altus Metrum

system, this can be accomplished in a TeleMega, TeleMetrum or TeleMini equipped rocket with less work than you may be accustomed to with other systems. It can even be fun!

Just prep the rocket for flight, then power up the altimeter in “idle” mode (placing air-frame horizontal for TeleMetrum or TeleMega, or selecting the Configure Altimeter tab for TeleMini). This will cause the firmware to go into “idle” mode, in which the normal flight state machine is disabled and charges will not fire without manual command. 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.5. Radio Link

TeleMetrum, TeleMini and TeleMega all incorporate an RF transceiver, but it’s not a full duplex system; each end can only be transmitting or receiving at any given moment. So we had to decide how to manage the link.

By design, the altimeter firmware listens for the radio link when it’s in “idle mode”, which allows us to use the radio link to configure the rocket, do things like ejection tests, and extract data after a flight without having to crack open the air-frame. However, when the board is in “flight mode”, the altimeter only transmits and doesn’t listen at all. That’s because we want to put ultimate priority on event detection and getting telemetry out of the rocket through the radio in case the rocket crashes and we aren’t able to extract data later.

We don’t generally use a 'normal packet radio' mode like APRS because they’re just too inefficient. The GFSK modulation we use is FSK with the base-band pulses passed through a Gaussian filter before they go into the modulator to limit the transmitted bandwidth. When combined with forward error correction and interleaving, this allows us to have a very robust 19.2 kilobit data link with only 10-40 milliwatts of transmit power, a whip antenna in the rocket, and a hand-held Yagi on the ground. We’ve had flights to above 21k feet AGL with great reception, and calculations suggest we should be good to well over 40k feet AGL with a 5-element yagi on the ground with our 10mW units and over 100k feet AGL with the 40mW devices. We hope to fly boards to higher altitudes over time, and would of course appreciate customer feedback on performance in higher altitude flights!

## A.6. APRS

TeleMetrum v2 and newer and TeleMega can send APRS if desired, and the interval between APRS packets can be configured. As each APRS packet takes a full second to transmit, we recommend an interval of at least 5 seconds to avoid consuming too much battery power or radio channel bandwidth. You can configure the time within each minute that APRS transmits by changing the APRS offset value. When the GPS signal is locked and knows the current time, the APRS offset selects the time with each minute for the first APRS transmission; subsequent transmissions occur each APRS interval seconds thereafter. You can configure the APRS interval and APRS offset using AltosUI; that process is described in [Configure Altimeter](#).

AltOS supports both compressed and uncompressed APRS position report data formats. The compressed format provides for higher position precision and shorter packets than the uncompressed APRS format. We’ve found some older APRS receivers that do not handle the

compressed format. The Kenwood TH-72A requires the use of uncompressed format to display altitude information correctly. The Yaesu FT1D requires the use of compressed format to display altitude information.

APRS packets include an SSID (Secondary Station Identifier) field that allows one operator to have multiple transmitters. AltOS allows you to set this to a single digit from 0 to 9, allowing you to fly multiple transmitters at the same time while keeping the identify of each one separate in the receiver. By default, the SSID is set to the last digit of the device serial number.

The APRS packet format includes a comment field that can have arbitrary text in it. AltOS uses this to send status information as shown in the following table.

Table 10. Altus Metrum APRS Comments

Field	Example	Description
1	L	GPS Status U for unlocked, L for locked
2	6	Number of Satellites in View
3	B4.0	Altimeter Battery Voltage
4	A3.7	Apogee Igniter Voltage
5	M3.7	Main Igniter Voltage
6	1286	Device Serial Number
4	1286	Device Serial Number

Here's an example of an APRS comment showing GPS lock with 6 satellites in view, a primary battery at 4.0V, and apogee and main igniters both at 3.7V from device 1286.

```
L6 B4.0 A3.7 M3.7 1286
```

Here's an example of an APRS comment showing GPS lock with 6 satellites in view and a primary battery at 4.0V from device 1876.

```
L6 B4.0 1876
```

Make sure your primary battery is above 3.8V any connected igniters are above 3.5V and GPS is locked with at least 5 or 6 satellites in view before flying. If GPS is switching between L and U regularly, then it doesn't have a good lock and you should wait until it becomes stable.

If the GPS receiver loses lock, the APRS data transmitted will contain the last position for which GPS lock was available. You can tell that this has happened by noticing that the GPS status character switches from 'L' to 'U'. Before GPS has locked, APRS will transmit zero for latitude, longitude and altitude.

## A.7. 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: Example Pyro Channel Configurations

Programming configurable pyro channels on Altus Metrum products that include them isn't difficult, but in an attempt to aid understanding of the configuration interface and help "keep simple things simple", we offer the following examples of the simplest configurations for common situations, along with some hints on avoiding unexpected results.

The rich set of conditions provided can be used to configure almost any pyro event you can imagine, for a wide variety of objectives. But don't be fooled! Typical events need only one or a few simple conditions to be configured for success. A key thing to remember is that **all** configured conditions must be true to allow a pyro channel to fire. Trying to include too many conditions often results in conflicting rules that never allow a channel to fire. The most important advice we can offer is, therefore, to try and find the simplest set of conditions that will do what you need for a given project.

## B.1. Two-Stage Flights

Successful completion of a two-stage flight often involves programming of two events. The first is firing a separation charge, the second is igniting the sustainer's (primary) motor.

Separation charges are best fired as soon as possible after the previous stage has completed providing acceleration, to minimize drag of the sustainer's coast phase before ignition. Recovery, whether the remainder of the flight is nominal or not, usually works best when the stages are separated. So, the "best" way to configure a pyro channel for a separation charge is to just set "after motor number". For a 2-stage project, set this to "1". This will cause the pyro channel to fire as soon as the firmware's flight state machine determines the first motor has burned out.

Safe ignition of a sustainer (primary) motor requires that it happen after the previous stage burns out, while the airframe remains mostly vertical, and typically after the sustainer has coasted away from the booster a bit. A good starting point is thus "after motor number" set the same as the separation charge, which is "1" for a 2-stage rocket. Then "angle from vertical less than" set to some reasonably vertical amount, perhaps 20 degrees. Then "delay after other conditions" set for the desired duration of coast. Use simulations to figure out what a reasonable value here is, but for typical high power rocketry sport flights that aren't trying to set records, something like 2 seconds is usually a good place to start.

## B.2. Triggered Clusters and Air Starts

When an airframe has a cluster of motors, one of which is "primary" and centered, surrounding by a ring of "secondary" motors, you may want to use the launch control system to fire the primary motor and use onboard electronics to light the rest of the cluster as soon as launch is detected. This is particularly true if the primary motor is significantly different in geometry and may take longer to come up to pressure than the secondary motors. In this case, a simple configuration to light secondary motors is "time since boost greater than" enabled and set to "0". There's really no point in setting an angle limit since no time has transpired for the airframe to change orientation.

Air starts can use the same simple configuration, but with the time set to a non-zero value. However, if air starts are going to light after the airframe leaves the launch rail or tower, add an "angle from vertical less than" condition just you would for a 2-stage sustainer to stay safe.

### B.3. Redundant Apogee

When flying a board like TeleMega or EasyMega, it's easy to configure a programmable channel to fire a redundant apogee charge. This is of course not **fully** redundant, since it's always possible that the board itself or its battery could be the failure source, but far more often, pyro events fail due to broken wires, bad connectors, or bad e-matches... so firing two charges from one board can add useful redundancy.

The simplest configuration for redundant apogee is "flight state after" set to "drogue", and then "delay after other conditions" set to a second or two.

### B.4. Redundant Main

Similarly to apogee, configuring a redundant main charge can provide useful redundancy. What we want is to configure an altitude for deployment lower than the primary main deploy altitude, and then ensure we only trigger on that condition while descending.

The simplest configuration for redundant main is "flight state after" set to "drogue", which will ensure we're in to the descent phase, then "height less than" set to a number lower than you've chosen for the primary main channel deployment height.

### B.5. Apogee Above Baro Sensor Limit

A question we've seen increasingly often is "How does the Telemega/Easymega detect apogee for flights above 100,000 feet?" Flights above that height are a bit outside our original design envelope, but can be made to work... This is **not** a simple flight, and the configuration for it is also not simple, but we think including this information is important for anyone contemplating such a project with our electronics!

Our flight computers use a Kalman sensor-fusing filter to estimate the flight state, which consists of three values:

1. Height above ground
2. Vertical speed
3. Vertical acceleration

Apogee is assumed to be where vertical speed crosses zero.

Below 30km altitude (about 100k'), we use both the barometer and the accelerometer to update the flight state, along with a basic Newtonian model of motion. That works well, pegging apogee within a few sensor samples essentially every time.

Above 30km, the barometric sensor doesn't provide useful data, so we can't use it to update the flight state. Instead, the Kalman filter falls back to a single sensor mode, using only the

accelerometer.

At all altitudes, we de-sense the barometric data when we estimate the speed is near or above mach as the sensor is often subjected to significant transients, which would otherwise push the flight state estimates too fast and could trigger a false apogee event.

That means the filter is no longer getting the benefit of two sensors, and relies on just the accelerometer. The trouble with accelerometers is they're measuring the derivative of speed, so you have to integrate their values to compute speed. Any offset error in acceleration measurement gets constantly added to that speed.

In addition, we assume the axial acceleration is actually vertical acceleration; our tilt measurements have enough integration error during coast that we can't usefully use that to get vertical acceleration. Because we don't live in an inertial frame, that means we're mis-computing the total acceleration acting on the airframe as we have to add gravity into the mix, and simply adding that to the axial acceleration value doesn't generate the right value.

The effect of this is to under-estimate apogee when you base the computation purely on acceleration as the rocket flies a parabolic path.

For flights **near** 100k', all of this works pretty well - you've got the flight state estimates adjusted using the barometric sensor up to 30km, then you're flying on inertial data to apogee.

For flights well above 100k', it's not great; you're usually going fast enough through 100k' that the baro sensor is still de-sensed through the end of its useful range, so the flight state estimates are not as close. After that, as you're flying purely on accelerometer data, there's no way to re-correct the state, so the apogee estimates can be off by quite a bit.

In the worst cases we have seen, the baro sensor data was wildly incorrect above mach due to poor static port design, leaving the state estimate of speed across the 30km boundary way off and causing the apogee detection to happen far from the correct time.

The good news is that correctly determining apogee is not really all that important at high altitudes; there's so little density that a drogue will have almost no drag anyways. Data from customer flights shows a very parabolic path down to about 50-60k feet, even with a recovery system deployed.

So, what we recommend is to set up two apogee plans:

1. Use the built-in apogee detection, but add a significant delay (as much as 30 seconds). This will probably fire near enough to apogee to not have a significant impact on the maximum height achieved.
2. Add a back-up apogee which fires after apogee **when the height is below about 20-25km**. This way, if the flight isn't nominal, and the sustainer ends up reaching apogee in dense air, you aren't hoping the chutes come out before it gets going too fast. And, you get a second pyro channel firing at that altitude even if it reached a higher altitude before.

You can wire these two pyro channels to the same pyro device; you just need to make sure they're wired + to + and - to - (the manual shows which screw terminals are which).

The bottom line is that flights to altitudes modestly above the range of the baro sensor with Altus Metrum products can be accomplished safely, but flying "way high" (like 300k') demands a deployment mechanism which doesn't solely rely on altimeters (like ours) which are designed for modest altitude rocketry. Flights to those altitudes also probably need active stabilization to make sure they follow the prescribed trajectory and stay inside their waiver.

# Appendix C: 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.



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 D: Updating Device Firmware

TeleMega, TeleMetrum v2 and newer, EasyMega, EasyMini and TeleDongle v3 are all programmed directly over their USB connectors (self programming). TeleMetrum v1, TeleMini v1.0 and TeleDongle v0.2 are all programmed by using another device as a programmer (pair programming). It's important to recognize which kind of devices you have before trying to reprogram them.

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/>

## D.1. Updating TeleMega, TeleMetrum v2 or newer, TeleMini v3, EasyMega, EasyMini, TeleDongle v3 or TeleBT v3 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.

### D.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.

### TeleMega

Connect pin 6 and pin 1 of the companion connector. Pin 1 can be identified by the square pad around it, and then the pins count sequentially across the board. Be very careful to **not** short pin 8 to anything as that is connected directly to the battery. Pin 7 carries 3.3V and the board will crash if that is connected to pin 1, but shouldn't damage the board.

### EasyMega

Connect pin 6 and pin 1 of the companion connector. Pin 1 can be identified by the square pad around it, and then the pins count sequentially across the board. Be very careful to **not** short pin 8 to anything as that is connected directly to the battery. Pin 7 carries 3.3V and the board will crash if that is connected to pin 1, but shouldn't damage the board.

### EasyTimer

Connect pin 5 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 5 the one on the other end of the row.

### TeleMetrum v2 and newer

Connect pin 6 and pin 1 of the companion connector. Pin 1 can be identified by the square pad around it, and then the pins count sequentially across the board. Be very careful to **not** short pin 8 to anything as that is connected directly to the battery. Pin 7 carries 3.3V and the board will crash if that is connected to pin 1, but shouldn't damage the board.

### 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.

### TeleDongle v3

Connect pin 32 on the CPU to ground. Pin 32 is closest to the USB wires on the row of pins towards the center of the board. Ground is available on the capacitor next to it, on the end towards the USB wires.

### TeleBT v3

Connect pin 4 on the CPU to 3.3V, which can be done by connecting pin 1 to pin 4. Pin 1 is the left pin on the lower edge of the chip when the unit is oriented such that the SMA is at the top.

## TeleBT v4

Connect pin 30 on the CPU to ground. Pin 30 is the 6th pin from the right on the top when the unit is oriented such that the SMA is at the top.

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.

## D.2. Pair Programming

The big concept to understand is that you have to use a TeleMetrum v1.0, TeleBT v1.0 or TeleDongle v0.2 as a programmer to update a pair programmed device. Due to limited memory resources in the cc1111, we don't support programming directly over USB for these devices.

If you need to update the firmware on a TeleDongle v0.2, we recommend updating the altimeter first, before updating TeleDongle. However, note that TeleDongle rarely need to be updated. Any firmware version 1.0.1 or later will work, version 1.2.1 may have improved receiver performance slightly.

### D.2.1. Updating TeleMetrum v1.x Firmware

1. Find the 'programming cable' that you got as part of the starter kit, that has a red 8-pin MicroMaTch connector on one end and a red 4-pin MicroMaTch connector on the other end.
2. Take the 2 screws out of the TeleDongle v0.2 or TeleBT v1.0 case to get access to the circuit board.
3. Plug the 8-pin end of the programming cable to the matching connector on the TeleDongle v0.2 or TeleBT v1.0, and the 4-pin end to the matching connector on the TeleMetrum. Note that each MicroMaTch connector has an alignment pin that goes through a hole in the PC board when you have the cable oriented correctly.
4. Attach a battery to the TeleMetrum board.
5. Plug the TeleDongle v0.2 or TeleBT v1.0 into your computer's USB port, and power up the TeleMetrum.
6. Run AltosUI, and select 'Flash Image' from the File menu.
7. Pick the TeleDongle v0.2 or TeleBT v1.0 device from the list, identifying it as the programming device.
8. Select the image you want put on the TeleMetrum, which should have a name in the form

telemetrum-v1.2-1.0.0.ihx. It should be visible in the default directory, if not you may have to poke around your system to find it.

9. 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.
10. Hit the 'OK' button and the software should proceed to flash the TeleMetrum with new firmware, showing a progress bar.
11. Confirm that the TeleMetrum board seems to have updated OK, which you can do by plugging in to it over USB and using a terminal program to connect to the board and issue the 'v' command to check the version, etc.

If something goes wrong, give it another try.

### D.2.2. Updating TeleMini v1.0 Firmware

You'll need a special 'programming cable' to reprogram the TeleMini v1.0. You can make your own using an 8-pin MicroMaTch connector on one end and a set of four pins on the other.

1. Take the 2 screws out of the TeleDongle v0.2 or TeleBT v1.0 case to get access to the circuit board.
2. Plug the 8-pin end of the programming cable to the matching connector on the TeleDongle v0.2 or TeleBT v1.0, and the 4-pins into the holes in the TeleMini v1.0 circuit board. Note that the MicroMaTch connector has an alignment pin that goes through a hole in the PC board when you have the cable oriented correctly, and that pin 1 on the TeleMini v1.0 board is marked with a square pad while the other pins have round pads.
3. Attach a battery to the TeleMini v1.0 board.
4. Plug the TeleDongle v0.2 or TeleBT v1.0 into your computer's USB port, and power up the TeleMini v1.0
5. Run AltosUI, and select 'Flash Image' from the File menu.
6. Pick the TeleDongle v0.2 or TeleBT v1.0 device from the list, identifying it as the programming device.
7. Select the image you want put on the TeleMini v1.0, which should have a name in the form telemini-v1.0-1.0.0.ihx. It should be visible in the default directory, if not you may have to poke around your system to find it.
8. 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.
9. Hit the 'OK' button and the software should proceed to flash the TeleMini v1.0 with new firmware, showing a progress bar.
10. Confirm that the TeleMini v1.0 board seems to have updated OK, which you can do by configuring it over the radio link through the TeleDongle, or letting it come up in "flight" mode and listening for telemetry.

If something goes wrong, give it another try.

### D.2.3. Updating TeleDongle v0.2 Firmware

Updating TeleDongle v0.2 firmware is just like updating TeleMetrum v1.x or TeleMini v1.0 firmware, but you use either a TeleMetrum v1.x, TeleDongle v0.2 or TeleBT v1.0 as the programmer.

1. Find the 'programming cable' that you got as part of the starter kit, that has a red 8-pin MicroMaTch connector on one end and a red 4-pin MicroMaTch connector on the other end.
2. Find the USB cable that you got as part of the starter kit, and plug the "mini" end in to the mating connector on TeleMetrum v1.x, TeleDongle v0.2 or TeleBT v1.0.
3. Take the 2 screws out of the TeleDongle v0.2 or TeleBT v1.0 case to get access to the circuit board.
4. Plug the 8-pin end of the programming cable to the matching connector on the programmer, and the 4-pin end to the matching connector on the TeleDongle v0.2. Note that each MicroMaTch connector has an alignment pin that goes through a hole in the PC board when you have the cable oriented correctly.
5. Attach a battery to the TeleMetrum v1.x board if you're using one.
6. Plug both the programmer and the TeleDongle into your computer's USB ports, and power up the programmer.
7. Run AltosUI, and select 'Flash Image' from the File menu.
8. Pick the programmer device from the list, identifying it as the programming device.
9. Select the image you want put on the TeleDongle v0.2, which should have a name in the form teledongle-v0.2-1.0.0.ihx. It should be visible in the default directory, if not you may have to poke around your system to find it.
10. 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. The TeleDongle v0.2 serial number is on the "bottom" of the circuit board, and can usually be read through the translucent blue plastic case without needing to remove the board from the case.
11. Hit the 'OK' button and the software should proceed to flash the TeleDongle v0.2 with new firmware, showing a progress bar.
12. Confirm that the TeleDongle v0.2 board seems to have updated OK, which you can do by plugging in to it over USB and using a terminal program to connect to the board and issue the 'v' command to check the version, etc. Once you're happy, remove the programming cable and put the cover back on the TeleDongle v0.2.

If something goes wrong, give it another try.

Be careful removing the programming cable from the locking 8-pin connector on TeleMetrum. You'll need a fingernail or perhaps a thin screwdriver or knife blade to gently pry the locking ears out slightly to extract the connector. We used a locking connector on TeleMetrum to help ensure that the cabling to companion boards used in a rocket don't ever come loose accidentally in flight.

# Appendix E: Frequency Calibration

All products that have radio interfaces require calibration of the radio frequency. Normally, this calibration is done once during the production process and the resulting cal value is saved into non-volatile memory. The procedure described here should only be used outside of the factory if you are really convinced the radio calibration is bad, and you have access to the required tools to do the calibration.

Because this procedure is only rarely needed in the field, we have not written any fancy user interface for doing it .. some interaction with and careful typing in a command-like style interface are required!

## E.1. Background Information

The radio system on each board uses a quartz crystal to control a frequency synthesizer that can be programmed to a range of operating frequencies. While these crystals are very stable, they have an accuracy specification that means once the base frequency they set is multiplied up to the typical operating range of our products, any variation also gets multiplied. The objective of the calibration process is, indirectly, to measure the actual operating frequency of the crystal and adjust the way the frequency synthesizer is programmed to account for this variation.

The frequency may shift a few tens of Hz over the full operating temperature range, and it may also shift a bit over time as the crystal ages. But once properly calibrated, none of those changes are likely to ever cause any operational problem, as the shift in operating frequency due to these factors is tiny compared to the bandwidth of our transmitted signal.

## E.2. Required Equipment

The calibration process requires the ability to precisely measure the actual frequency of a steady CW carrier on or about the intended operating frequency in the vicinity of 435 MHz.

In production, we use an HP 5385A that is locked to a 10 MHz reference that is in turn locked to GPS, which provides a highly accurate calibration. Any reasonably accurate frequency counter is likely to be sufficient.

You also need a computer with terminal program and USB cable to attach to the board in question, along with a battery and power switch suitable for powering the board up.

## E.3. RF Calibration Procedure

Using the terminal program, connect to the board over USB. You will find that you are now interacting with a command interpreter on the board. Using '?' will show the available commands. Of interest for this process are the 'C' command which turns on a steady transmitted carrier on the currently selected operating frequency, and the 'c' subcommands that allow interaction with the saved configuration.

Use the 'c s' command to discover and note the current radio calibration value, and the

operating frequency the board is configured for in kHz.

Set up your frequency counter with a suitable antenna near the board's antenna and use the 'C' command to turn on a steady carrier. Let the frequency stabilize, and note what it is to as many digits as are steady on your counter's display.

To calculate the new calibration value, the equation is:

1.  $(\text{intended\_frequency} / \text{measured\_frequency}) * \text{current\_cal\_value}$

Set the new calibration value using 'c f <value>', then use 'c w' to save that cal value into non-volatile memory. You can use the 'C' command again to confirm the operating frequency is now within a few 10's of Hz of the intended operating frequency.

# Appendix F: Flight Data Recording

Each flight computer logs data at 100 samples per second during ascent and 10 samples per second during descent, except for TeleMini v1.0, which records ascent at 10 samples per second and descent at 1 sample per second. Data are logged to an on-board flash memory part, which can be partitioned into several equal-sized blocks, one for each flight.

Table 11. Data Storage on Altus Metrum altimeters

Device	Bytes per Sample	Total Storage	Minutes at Full Rate
TeleMetrum v1.0	8	1MB	20
TeleMetrum v1.1 v1.2	8	2MB	40
TeleMetrum v2.0	16	8MB	80
TeleMetrum v3.0	16	8MB	80
TeleMini v1.0	2	5kB	4
TeleMini v3.0	16	512kB	5
EasyMini	16	1MB	10
TeleMega	32	8MB	40
EasyMega	32	8MB	40

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 TeleMetrum v1.x, TeleMini v3.0 and EasyMini. This consumes 64kB of flash space. This configuration space is not available for storing flight log data.

TeleMetrum v2 or newer, TeleMega and EasyMega store configuration data in a bit of eeprom available within the processor chip, leaving that space available in flash for more flight 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 underestimate the storage (in bytes) needed for the flight. For instance, a TeleMetrum v2 or newer flight spending 20 seconds in ascent and 150 seconds in descent will take about  $(20 * 1600) + (150 * 160) = 56000$  bytes of storage. You could store dozens of these flights in the on-board flash.

The default size allows for several flights on each flight computer, except for TeleMini v1.0, which only holds data for a single flight. 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 G: Hardware Specifications

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

Table 12. Altus Metrum Flight Computer Electronics

Device	Barometer	Z-axis accel	GPS	3D sensors	Storage	RF Output	Battery	Battery Charger
TeleMetrum v1.0	MP3H6115 10km (33k')	MMA2202 50g	SkyTraq	-	1MB	10mW	3.7V	MAX1555 100mA
TeleMetrum v1.1	MP3H6115 10km (33k')	MMA2202 50g	SkyTraq	-	2MB	10mW	3.7V	MAX1555 100mA
TeleMetrum v1.2	MP3H6115 10km (33k')	ADXL78 70g	SkyTraq	-	2MB	10mW	3.7V	MAX1555 100mA
TeleMetrum v2.0	MS5607 30km (100k')	MMA6555 102g	uBlox Max-7Q	-	8MB	40mW	3.7V	MCP73831/2 500mA
TeleMetrum v3.0	MS5607 30km (100k')	ADXL375 200g	uBlox Max-8Q	-	8MB	40mW	3.7V	MCP73831/2 400mA
TeleMetrum v4.0	MS5607 30km (100k')	ADXL375 200g	uBlox Max-8C/10S	-	8MB	40mW	3.7V	MCP73831/2 400mA
TeleMini v1.0	MP3H6115 10km (33k')	-	-	-	5kB	10mW	3.7V	-
TeleMini v3.0	MS5607 30km (100k')	-	-	-	512kB	40mW	3.7V	-
EasyMini v1.0-v3.0	MS5607 30km (100k')	-	-	-	1MB	-	3.7-12V	-
TeleMega v1.0	MS5607 30km (100k')	MMA6555 102g	uBlox Max-7Q	MPU6000 HMC5883	8MB	40mW	3.7V	MCP73831/2 500mA
TeleMega v2.0	MS5607 30km (100k')	MMA6555 102g	uBlox Max-7Q	MPU6000 HMC5883	8MB	40mW	3.7V	MCP73831/2 500mA
TeleMega v3.0	MS5607 30km (100k')	MMA6555 102g	uBlox Max-7Q	MPU9250	8MB	40mW	3.7V	MCP73831/2 500mA
TeleMega v4.0	MS5607 30km (100k')	ADXL375 200g	uBlox Max-8Q	BMX160	8MB	40mW	3.7V	MCP73831/2 400mA
TeleMega v5.0	MS5607 30km (100k')	ADXL375 200g	uBlox Max-8Q	MPU6000 MMC5983	8MB	40mW	3.7V	MCP73831/2 400mA
TeleMega v6.0	MS5607 30km (100k')	ADXL375 200g	uBlox Max-8Q	BMI088 MMC5983	8MB	40mW	3.7V	MCP73831/2 400mA
TeleMega v7.0	MS5607 30km (100k')	ADXL375 200g	uBlox Max-10S	BMI088 MMC5983	8MB	40mW	3.7V	MCP73831/2 400mA
EasyMega v1.0	MS5607 30km (100k')	MMA6555 102g	-	MPU6000 HMC5883	8MB	-	3.7V	MCP73831/2 500mA
EasyMega v2.0	MS5607 30km (100k')	ADXL375 200g	-	MPU9250	8MB	-	3.7V	MCP73831/2 400mA
EasyMega v3.0	MS5607 30km (100k')	ADXL375 200g	-	BMI088 MMC5983A	8MB	-	3.7V	MCP73831/2 400mA
TeleGPS v1.0	-	-	uBlox Max-7Q	-	2MB	10mW	3.7V	MCP73831/2 500mA
TeleGPS v2.0	-	-	uBlox Max-7Q	-	2MB	40mW	3.7V	MCP73831/2 400mA
TeleGPS v3.0	-	-	uBlox Max-10S	-	2MB	40mW	3.7V	MCP73831/2 400mA
TeleGPS v4.0	-	-	uBlox Max-10S	-	2MB	40mW	3.7V	MCP73831/2 400mA

Device	Barometer	Z-axis accel	GPS	3D sensors	Storage	RF Output	Battery	Battery Charger
EasyTimer v1.0	-	16g	-	BMX160	-	-	3.7-12V	-
EasyTimer v2.0	-	24g	-	BMI088 MMC5983	1MB	-	3.7-12V	-
EasyMotor v3.0	-	ADXL375 200g	-	-	8MB	-	3.7V	MCP73831/2 400mA

Table 13. Altus Metrum Flight Computer Mechanical Components

Device	Connectors	Screw Terminals	Width	Length	Tube Size
TeleMetrum	Antenna Debug Companion USB Battery	Apogee pyro Main pyro Switch	1 inch (2.54cm)	2 ¾ inch (6.99cm)	29mm coupler
TeleMini v1.0	Antenna Debug Battery	Apogee pyro Main pyro	½ inch (1.27cm)	1½ inch (3.81cm)	18mm coupler
TeleMini v2.0	Antenna Debug USB Battery	Apogee pyro Main pyro Battery Switch	½ inch (1.27cm)	1.67 inch (4.24cm)	18mm coupler
EasyMini	Debug USB Battery	Apogee pyro Main pyro Battery	0.8 inch (2.03cm)	1½ inch (3.81cm)	24mm coupler
TeleMega	Antenna Debug Companion USB Battery	Apogee pyro Main pyro Pyro A-D Switch Pyro battery	1¼ inch (3.18cm)	3¼ inch (8.26cm)	38mm coupler
EasyMega	Debug Companion USB Battery	Apogee pyro Main pyro Pyro A-D Switch Pyro battery	1¼ inch (3.18cm)	2¼ inch (5.62cm)	38mm coupler
EasyTimer	Debug USB Battery	Pyro A Pyro B Battery	0.8 inch (2.03cm)	1½ inch (3.81cm)	24mm coupler
EasyMotor	Debug USB	+5V Pres GND Switch Battery	0.8 inch (2.03cm)	1½ inch (3.81cm)	24mm coupler

# Appendix H: Release Notes

## H.1. Release Notes for Version 1.9.22

Version 1.9.22

### H.1.1. AltOS

- Add support for external Mosaic GPS receiver for TeleMega

### H.1.2. AltosUI & TeleGPS application

- Support TeleMega v7 .eeprom files
- Add external Mosaic GPS selection option in Altimeter configuration

## H.2. Release Notes for Version 1.9.21

Version 1.9.21

### H.2.1. AltOS

- Add support for TeleBT v4.0 factory test.
- Support Picolibc 1.8.10

### H.2.2. AltosUI & TeleGPS application

- Support pressure values below sensor range.

## H.3. Release Notes for Version 1.9.20

Version 1.9.20

### H.3.1. AltOS

- Add support for TeleGPS v4.0 and TeleMega v7.0

### H.3.2. AltosUI & TeleGPS application

- Add support for TeleGPS v4.0 and TeleMega v7.0
- Add support for 30V pyro voltage range

## H.4. Release Notes for Version 1.9.19

Version 1.9.19

### H.4.1. AltOS

- Finish support for EasyMega V3.

### H.4.2. AltosUI & TeleGPS application

- Show device configuration information when graphing a flight.

## H.5. Release Notes for Version 1.9.18

Version 1.9.18

### H.5.1. AltOS

- Add support for EasyTimer V2. The new version of this product has on-board storage to log data during flight.

### H.5.2. AltosUI & TeleGPS application

- Add support for EasyTimer V2. This includes support for analyzing flight data from the on-board logs.
- Allow on-board beepers to be disabled by setting the frequency to 0.

## H.6. Release Notes for Version 1.9.17

Version 1.9.17

### H.6.1. AltOS

- Fix TeleMini v3 Monitor Idle support
- Support TeleMetrum v4.0 with uBlox-10 GPS module
- Improve igniter reporting via the beeper.

### H.6.2. AltosUI & TeleGPS application

- Add support for EasyMini v3 Monitor Idle

## H.7. Release Notes for Version 1.9.16

Version 1.9.16

### H.7.1. AltOS

- Add TeleGPS v3.0 support

### H.7.2. AltosUI & TeleGPS application

- Add TeleGPS v3.0 support

## H.8. Release Notes for Version 1.9.15

Version 1.9.15

### H.8.1. AltOS

- Add TeleMega v6.0 support
- Add TeleMetrum v4.0 support
- Fix sign of IMU values for TeleMega v5 boards in the 'across' axis. This affects IMU acceleration and gyro reports for that axis, but has no effect on in-flight operation of the tilt computation.

## H.9. Release Notes for Version 1.9.14

Version 1.9.14

### H.9.1. AltOS

- Fix 1.9.13 regression in TeleLCO startup sequence that detects available TeleFire units.

# H.10. Release Notes for Version 1.9.13

Version 1.9.13

## H.10.1. AltOS

- Add option to beep max height in feet after landing
- Fix APRS reports to be sent at the correct time and spacing.
- Fix possible barometric sensor communication failure when the CPU is busy talking to the radio at the same time. This would cause loss of telemetry and failure to track the state of the rocket during flight. This was aggravated by the APRS reports getting sent more often than they should.
- Change EasyMotor v3 code to base logging on motor pressure rather than the accelerometer. This allows use of EasyMotor v3 in a static test stand.

## H.10.2. AltosUI

- Add support for configuring the units used to report height after landing on the beeper.

# H.11. Release Notes for Version 1.9.12

Version 1.9.12

## H.11.1. AltOS

- Add EasyMini v3.0 and EasyMotor v3.0 support
- Fix TeleMetrum v2.0 configuration. Saving config would crash the board.

## H.11.2. AltosUI

- Add EasyMotor log parsing and graphing.

## H.12. Release Notes for Version 1.9.11

Version 1.9.11

### H.12.1. AltOS

- Make Apogee Delay work again.
- Allow TX power to be limited to 10mW for compliance with some uses under UK regulations.
- Fix numerous minor issues with 16- vs 32- bit time values.

### H.12.2. AltosUI

- Support M1-based Macs, follow AdoptOpenJDK to Adoptium

### H.12.3. AltosDroid

- Handle Bluetooth permissions reliably.
- Fix some screen rotation bugs.

## H.13. Release Notes for Version 1.9.10

Version 1.9.10

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

### H.13.1. AltosUI

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

### H.13.2. 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.

## H.14. 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.

### H.14.1. 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 calibration, 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.

### H.14.2. 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.

# H.15. Release Notes for Version 1.9.8

Version 1.9.8

## H.15.1. AltOS

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

## H.15.2. AltosUI

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

## H.15.3. AltosDroid

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

## H.15.4. MicroPeak GUI

- Show error dialog if device open fails

# H.16. Release Notes for Version 1.9.7

Version 1.9.7

## H.16.1. AltOS

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

## H.16.2. 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

## H.16.3. AltosDroid

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

## H.17. Release Notes for Version 1.9.6

Version 1.9.6

### H.17.1. 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.

# H.18. Release Notes for Version 1.9.5

Version 1.9.5

## H.18.1. 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.

## H.18.2. 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.

## H.19. Release Notes for Version 1.9.4

Version 1.9.4

### H.19.1. 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.

### H.19.2. AltosUI

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

## H.20. Release Notes for Version 1.9.3

Version 1.9.3

### H.20.1. 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.

### H.20.2. 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.

## H.21. Release Notes for Version 1.9.2

Version 1.9.2

### H.21.1. AltOS

- Add support for TeleMega v4.0
- Fix time wrapping issue with TeleLCO and TeleFire

### H.21.2. AltosUI, TeleGPS, MicroPeak

- Add column in AltosUI for IMU data from TeleMega and EasyMega

### H.21.3. AltosDroid

- Allow sorting of trackers by call, serial, frequency or age
- Offer selection of font sizes
- Various bug fixes for older Android versions

## H.22. Release Notes for Version 1.9.1

Version 1.9.1

### H.22.1. 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.

### H.22.2. 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.

## H.23. Release Notes for Version 1.9

Version 1.9

### H.23.1. AltOS

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

### H.23.2. AltosUI, TeleGPS, MicroPeak

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

## H.24. Release Notes for Version 1.8.7

Version 1.8.7

### H.24.1. AltOS

- Include TeleMega v3.0 firmware

### H.24.2. 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.

## H.25. Release Notes for Version 1.8.6

Version 1.8.6

### H.25.1. 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  $\geq$  instead of  $=$
- Change Time since Boost to be Time since launch.

### H.25.2. 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.

### H.25.3. MicroPeak

- Report altimeter-recorded maximum height value

## H.26. 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.

### H.26.1. AltOS

- Fix startup beeps that indicate sensor failures.

### H.26.2. AltosUI, TeleGPS

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

## H.27. Release Notes for Version 1.8.4

Version 1.8.4 includes support for EasyMini version 2.0

### H.27.1. AltOS

- Support for EasyMini version 2.0 hardware.

## H.28. 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.

### H.28.1. 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'.

### H.28.2. 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.

## H.29. 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.

### H.29.1. AltOS

AltOS New Features

- Support for TeleGPS version 2.0 hardware.

### H.29.2. 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.

## H.30. 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.

### H.30.1. 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.

### H.30.2. 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.

## H.31. 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.

### H.31.1. AltOS

AltOS New Features

- Add support for TeleBT v4.0 boards.

### H.31.2. 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.

## H.32. 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.

### H.32.1. 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.

### H.32.2. AltosUI and TeleGPS Applications

AltosUI New Features

- Add support for TeleMini v3.0 hardware

## H.33. 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.

### H.33.1. 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.

### H.33.2. 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.

## H.34. 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.

### H.34.1. 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.

### H.34.2. AltosUI, TeleGPS and AltosDroid Applications

AltosUI fixes:

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

## H.35. 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.

### H.35.1. 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.

### H.35.2. 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.

### H.35.3. Documentation

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

## H.36. 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 and android devices along with Bluetooth support for Windows.

### H.36.1. 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.

### H.36.2. 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.

### H.36.3. AltosDroid

AltosDroid new features:

- Monitor Idle mode. Check state of flight computer while in idle mode over the radio link
- Fire Igniters. Remotely fire igniters 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.

#### **H.36.4. Documentation**

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

## H.37. 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.

### H.37.1. 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.

### H.37.2. 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.

### H.37.3. 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

## H.38. Release Notes for Version 1.6.1

Version 1.6.1 includes support for our updated TeleBT v3.0 product and bug fixes in in the flight software for all our boards and ground station interfaces.

### H.38.1. AltOS

AltOS New Features:

- Add support for TeleBT v3.0 boards.
- Add support for uncompressed APRS data, providing support for older APRS receivers. Uncompressed APRS data is less precise, takes more bandwidth and doesn't have integrated altitude data.

AltOS Fixes:

- Make TeleDongle and TeleBT more tolerant of data rate variations from transmitting devices.

### H.38.2. AltosUI and TeleGPS Applications

AltosUI and TeleGPS New Features:

- Add map to Monitor Idle display. It's nice to be able to verify that maps are working, instead of needing to use Monitor Flight.

AltosUI and TeleGPS Fixes:

- Fix frequency configuration to round values instead of truncate them, avoiding a common 1 kHz error in the setting.
- Turn the Windows stub into a more useful program that can launch the application with parameters so that file manager icons work more reliably.
- Force KML export to use a C locale so that numbers are formatted with '.' instead of ',' for a decimal separator in non-US locales.
- Preload map tiles based on distance rather than number of tiles; this means you get the same resolution covering the entire area, rather than having high resolution near the center and low resolution further away.
- Allow configuration of frequency and callsign in Monitor Idle mode.
- Fix layout weirdness when resizing windows on Windows. Windows shouldn't have giant blank spaces around the useful content anymore.
- Fix layout weirdness when resizing windows on Windows. Windows shouldn't have giant blank spaces around the useful content anymore.
- Use a longer filter for descent speed values. This should provide something more useful on the display, although it will take longer to respond to changes now.
- Make Replay Flight run in realtime again. It had been set to run at 10x speed by mistake.

### H.38.3. AltosDroid

AltosDroid New Features:

- Add offline map support using mapping code from AltosUI.
- Support TeleDongle (and TeleBT via USB) on devices supporting USB On-The-Go.
- Display additional TeleMega pyro channel status in Pad tab.
- Switch between metric and imperial units.
- Monitor TeleBT battery voltage.
- Track multiple devices at the same time, selecting between them with a menu or using the map.
- Add hybrid, satellite and terrain map types.

AltosDroid Fixes:

- Use standard Android display conventions so that a menu button is available in the application title bar.
- Adjust layout to work on large and small screens; shrinking the go/no-go lights in smaller environments to try and make everything visible.
- Make voice announcements depend on current tab.
- Compute adjustment to current travel direction while in motion towards rocket.

## H.39. Release Notes for Version 1.6

Version 1.6 includes support for our updated TeleDongle v3.0 product and bug fixes in the flight software for all our boards and ground station interfaces.

### H.39.1. AltOS

#### AltOS New Features

- Add support for TeleDongle v3.0 boards.

#### AltOS Fixes

- Don't beep out the continuity twice by accident in idle mode. If the battery voltage report takes longer than the initialization sequence, the igniter continuity would get reported twice.
- Record all 32 bits of gyro calibration data in TeleMega and EasyMega log files. This fixes computation of the gyro rates in AltosUI.
- Change TeleDongle LED usage. Green LED flashes when valid packet is received. Red LED flashes when invalid packet is received.
- Replace LPC1114 SPI driver with non-interrupt version. The interrupt code would occasionally wedge on long transfers if interrupts were blocked for too long. This affects all released TeleGPS products; if you have a TeleGPS device, you'll want to reflash the firmware.

### H.39.2. AltosUI and TeleGPS Applications

#### AltosUI and TeleGPS New Features

- Compute tilt angle from TeleMega and EasyMega log files. This duplicates the quaternion-based angle tracking code from the flight firmware inside the ground station software so that post-flight analysis can include evaluation of the tilt angle.
- Shows the tool button window when starting with a data file specified. This means that opening a data file from the file manager will now bring up the main window to let you operate the whole application.

#### AltosUI Fixes

- Show the 'Connecting' dialog when using Monitor Idle. Lets you cancel the Monitor Idle startup when connecting over the radio link.
- Make 'Monitor Idle' work for TeleGPS devices when connected over USB. It's nice for testing without needing to broadcast over the radio.
- Use different Windows API to discover USB devices. This works better on my Windows 7 box, and will be used if the older API fails to provide the necessary information.
- Look in more places in the registry to try and identify the installed Java version on Windows. If you install the default 32-bit version of Windows on a 64-bit OS, the Java registry information is hiding \SOFTWARE\Wow6432Node for some reason.
- Fix file association on Windows by searching for the javaw.exe program instead of assuming

it is in %SYSTEMROOT%. This makes double-clicking on Altus Metrum data files in the file manager work correctly.

- When replaying a file, put 'done' in the Age field when we reach the end of the file, instead of continuing to count forever.
- In the Scan Channels code, wait for five seconds if we see any packet. This is needed because AltOS now sends the callsign, serial number and flight number only once every five seconds these days.
- In the Scan Channels code, reset pending flight state information each time we change channels. This avoids having flight computers appear on multiple frequencies by accident.

## H.40. Release Notes for Version 1.5

Version 1.5 is a major release. It includes support for our new EasyMega product, new features and bug fixes in in the flight software for all our boards and the AltosUI ground station

### H.40.1. AltOS

#### AltOS New Features

- Add support for EasyMega boards.
- Make the APRS SSID be configurable. This lets you track different rockets on the same receiver without getting things mixed up.
- Report extra pyro channel continuity state on EasyMega and TeleMega via the beeper. This lets you easily verify flight readiness on these boards after powering up the electronics on the rail.
- Add lower telemetry data rates (2400 and 9600 bps) to increase telemetry radio range. This reduces the amount of data received as well as increasing battery consumption in the transmitter.
- Change TeleGPS to have only a single log, and append new data to it rather than using separate per-flight logs. This avoids accidentally filling up log storage by turning TeleGPS on/off several times.

#### AltOS Fixes

- Increase the maximum range for altitude values from +/-32767m to +/-2147483647m, allowing the flight computers to function correctly above the 32km level.
- Continuously test pyro firing conditions during delay stage, inhibiting the pyro channel if the test fails. This prevents firing pyro charges where the conditions were good before the delay, but become bad before the delay expires.
- Allow negative numbers in pyro configuration values. This lets you specify things like descending speed or deceleration.

### H.40.2. AltosUI and TeleGPS Applications

#### AltosUI and TeleGPS New Features

- Support telemetry baud rate selection. Adds menus to the flight monitoring and configuration for baud rate selection.
- Support APRS SSID configuration.
- Integrate with file managers. This provides icons for all of our file types and associates our application with the files so that using a file manager to open a AltOS data file results in launching our application.

#### AltosUI Fixes

- Make the 'Graph' button on the landed tab work again.

- Make tests for Java on Windows a bit smarter, and also provide the user with the option to skip installing Java for cases where we just can't figure out what version is installed.

## H.41. Release Notes for Version 1.4.2

Version 1.4.2 is a minor release. It fixes Java-related install issues on Windows

### H.41.1. AltosUI and TeleGPS Applications

Windows Install Fixes

- Checks for Java installation data in more registry locations.
- Allows user to bypass Java installation in case the detection fails.

## H.42. Release Notes for Version 1.4.1

Version 1.4.1 is a minor release. It fixes install issues on Windows and provides the missing TeleMetrum V2.0 firmware. There aren't any changes to the firmware or host applications at all. All Windows users will want to upgrade to get the signed driver, but Mac and Linux users who do not need the TeleMetrum V2.0 firmware image will not need to upgrade.

### H.42.1. AltosUI and TeleGPS Applications:

#### Windows Install Fixes

- Provide signed Windows driver files. This should avoid any need to disable driver signature checking on Windows 7 or 8.
- Fix Java version detection and download. Previously, the installer would only look for Java 6 or 7 and insist on downloading its own Java bits if there was something else installed. Furthermore, the 64-bit Java link provided didn't work for anyone other than Keith, making it impossible to install AltOS on any machine with Java SE 8 installed.

#### Other Fixes

- Include 1.4 firmware for TeleMetrum V2.0. None of the installers shipped this file. Now it's included in the AltOS packages for Linux, Mac and Windows.
- Include Google Application Key for map downloading. The 1.4 release didn't have this key in the released version of the software, making map downloading fail for most people.

## H.43. Release Notes for Version 1.4

Version 1.4 is a major release. It includes support for our new TeleGPS product, new features and bug fixes in in the flight software for all our boards and the AltosUI ground station

### H.43.1. AltOS

AltOS new features:

- Add support for TeleGPS boards.
- Make the beeper tone configurable, making it possible to distinguish between two Altus Metrum products in the same ebay.
- Make the firing time for extra pyro channels configurable, allowing longer (or shorter) than the default 50ms. Only relevant for TeleMega at this time.

AltOS fixes:

- Replace the 'dit dit dit' tones at startup with the current battery voltage, measured in tenths of a volt. This lets you check the battery voltage without needing telemetry, which is especially useful on EasyMini.
- Change state beeping to "Farnsworth spacing", which means they're quite a bit faster than before, and so they take less time to send.
- Fix bug preventing the selection of the 'Flight State After' mode in pyro configuration.
- Fix bug where erasing flights would reset the flight number to 2 on TeleMega and TeleMetrum v2.
- Fix u-Blox GPS driver to mark course and speed data as being present.

### H.43.2. AltosUI Application

AltosUI new features:

- Add zooming and new content types (terrain and road maps) to map view. Change map storage format from PNG to Jpeg, which saves a huge amount of disk space. You will need to re-download all of your pre-loaded map images.
- Add a distance measuring device to the maps view. Select this by using any button other than the left one, or by pressing shift or control on the keyboard while using the left button.
- Add new 'Ignitor' tab to the flight monitor display for TeleMega's extra ignitors.
- Add additional ignitor firing marks and voltages to the graph so you can see when the ignitors fired, along with the ignitor voltages.
- Add GPS course, ground speed and climb rate as optional graph elements.

AltosUI fixes:

- When flashing new firmware, re-try opening the device as sometimes it takes a while for the underlying operating system to recognize that the device has rebooted in preparation for the

flashing operation.

- Hide Tilt Angle in ascent tab for devices that don't have a gyro.
- Increase the width of data lines in the graphs to make them easier to read.
- Filter out speed and acceleration spikes caused by ejection charge firing when computing the maximum values. This provides a more accurate reading of those maximums.
- Fix EasyMini voltage displays. Early EasyMini prototypes used a 3.0V regulator, and AltosUI still used that value as the basis of the computation. Production EasyMini boards have always shipped with a 3.3V regulator. Also, purple EasyMini boards sensed the battery voltage past the blocking diode, resulting in a drop of about 150mV from the true battery voltage. Compensate for that when displaying the value.
- Display error message when trying to configure maximum flight log size while the flight computer still has flight data stored.
- Handle TeleMetrum and TeleMini eeprom files generated with pre-1.0 firmware. Those ancient versions didn't report the log format, so just use the product name instead.

### H.43.3. TeleGPS Application

- New application designed for use with TeleGPS boards.
- Shares code with AltosUI, mostly just trimmed down to focus on TeleGPS-related functions.

### H.43.4. Documentation

Documentation changes:

- Re-create the drill template images; they should print correctly from Firefox at least. Ship these as individual PDF files so they're easy to print.
- Add a description of the 'Apogee Lockout' setting, which prevents the apogee charge from firing for a configurable amount of time after boost.

## H.44. Release Notes for Version 1.3.2

Version 1.3.2 is a minor release. It includes small bug fixes for the TeleMega flight software and AltosUI ground station

### H.44.1. AltOS

AltOS fixes:

- On TeleMega, limit number of logged GPS status information to 12 satellites. That's all there is room for in the log structure.
- Improve APRS behavior. Remembers last known GPS position and keeps sending that if we lose GPS lock. Marks locked/unlocked by sending L/U in the APRS comment field along with the number of sats in view and voltages.

### H.44.2. AltosUI Application

AltosUI fixes:

- If the TeleMega flight firmware reports that it has logged information about more than 12 satellites, don't believe it as the log only holds 12 satellite records.
- Track the maximum height as computed from GPS altitude data and report that in the flight summary data.
- Use letters (A, B, C, D) for alternate pyro channel names instead of numbers (0, 1, 2, 3) in the Fire Igniter dialog.

## H.45. Release Notes for Version 1.3.1

Version 1.3.1 is a minor release. It improves support for TeleMega, TeleMetrum v2.0, TeleMini v2.0 and EasyMini.

### H.45.1. AltOS

AltOS new features:

- Improved APRS mode. Now uses compressed position format for smaller data size, improved precision and to include altitude data as well as latitude and longitude. Also added battery and pyro voltage reports in the APRS comment field so you can confirm that the unit is ready for launch.

AltOS fixes:

- Improve sensor boot code. If sensors fail to self-test, the device will still boot up and check for pad/idle modes. If in idle mode, the device will warn the user with a distinct beep, if in Pad mode, the unit will operate as best it can. Also, the Z-axis accelerometer now uses the factory calibration values instead of re-calibrating on the pad each time. This avoids accidental boost detect when moving the device around while in Pad mode.
- Fix antenna-down mode accelerometer configuration. Antenna down mode wasn't working because the accelerometer calibration values were getting re-computed incorrectly in inverted mode.

### H.45.2. AltosUI Application

AltosUI new features:

- Display additional TeleMega sensor values in real units. Make all of these values available for plotting. Display TeleMega orientation value in the Ascent and Table tabs.
- Support additional TeleMega pyro channels in the Fire Igniter dialog. This lets you do remote testing of all of the channels, rather than just Apogee and Main.

AltosUI fixes:

- Limit data rate when downloading satellite images from Google to make sure we stay within their limits so that all of the map tiles download successfully.

## H.46. Release Notes for Version 1.3

Version 1.3 is a major release. It adds support for TeleMega, TeleMetrum v2.0, TeleMini v2.0 and EasyMini.

### H.46.1. AltOS

AltOS new features:

- Add STM32L processor support. This includes enhancements to the scheduler to support products with many threads.
- Add NXP LPC11U14 processor support.
- Support additional pyro channels. These are configurable through the UI to handle air starts, staging, additional recovery events and external devices such as cameras.
- Add 3-axis gyro support for orientation tracking. This integrates the gyros to compute the angle from vertical during flight, allowing the additional pyro events to be controlled by this value.
- Many more device drivers, including u-Blox Max 7Q GPS, Freescale MMA6555 digital single-axis accelerometer, Invensense MPU6000 3-axis accelerometer + 3 axis gyro, Honeywell HMC5883 3-axis magnetic sensor and the TI CC1120 and CC115L digital FM transceivers

### H.46.2. AltosUI Application

AltosUI new features:

- Support TeleMega, TeleMetrum v2.0, TeleMini v2.0 and EasyMini telemetry and log formats.

AltosUI fixes:

- Use preferred units for main deployment height configuration, instead of always doing configuration in meters. == MicroPeak Application
- Add 'Download' button to menu bar.
- Save the last log directory and offer that as the default for new downloads

## H.47. Release Notes for Version 1.2.1

Version 1.2.1 is a minor release. It adds support for TeleBT and the AltosDroid application, provides several new features in AltosUI and fixes some bugs in the AltOS firmware.

### H.47.1. AltOS

AltOS new features:

- Add support for TeleBT

AltOS fixes:

- In TeleMini recovery mode (when booted with the outer two debug pins connected together), the radio parameters are also set back to defaults (434.550MHz, NOCALL, factory radio cal).
- Correct Kalman filter model error covariance matrix. The values used previously assumed continuous measurements instead of discrete measurements.
- Fix some bugs in the USB driver for TeleMetrum and TeleDongle that affected Windows users.
- Adjusted the automatic gain control parameters that affect receive performance for TeleDongle. Field tests indicate that this may improve receive performance somewhat.

### H.47.2. AltosUI Application

AltosUI application new features:

- Make the initial position of the AltosUI top level window configurable. Along with this change, the other windows will pop up at 'sensible' places now, instead of on top of one another.
- Add GPS data and a map to the graph window. This lets you see a complete summary of the flight without needing to 'replay' the whole thing.

AltosUI application fixes:

- Handle missing GPS lock in 'Descent' tab. Previously, if the GPS position of the pad was unknown, an exception would be raised, breaking the Descent tab contents.
- Improve the graph, adding tool-tips to show values near the cursor and making the displayed set of values configurable, adding all of the flight data as options while leaving the default settings alone so that the graph starts by showing height, speed and acceleration.
- Add callsign to Monitor idle window and connecting dialogs. This makes it clear which callsign is being used so that the operator will be aware that it must match the flight computer value or no communication will work.
- When downloading flight data, display the block number so that the user has some sense of progress. Unfortunately, we don't know how many blocks will need to be downloaded, but at least it isn't just sitting there doing nothing for a long time.

### H.47.3. AltosDroid

- First version of this application

## H.48. Release Notes for Version 1.2

Version 1.2 is a major release. It adds support for MicroPeak and the MicroPeak USB adapter.

### H.48.1. AltOS

AltOS New Features:

- Add MicroPeak support. This includes support for the ATtiny85 processor and adaptations to the core code to allow for devices too small to run the multi-tasking scheduler.

### H.48.2. AltosUI and MicroPeak Application

New Features:

- Added MicroPeak application

AltosUI and MicroPeak fixes:

- Distribute Mac OS X packages in disk image ('.dmg') format to greatly simplify installation.
- Provide version numbers for the shared Java libraries to ensure that upgrades work properly, and to allow for multiple Altus Metrum software packages to be installed in the same directory at the same time.

## H.49. Release Notes for Version 1.1

Version 1.1.1 is a bug-fix release. It fixes a couple of bugs in AltosUI and one firmware bug that affects TeleMetrum version 1.0 boards. Thanks to Bob Brown for help diagnosing the Google Earth file export issue, and for suggesting the addition of the Ground Distance value in the Descent tab.

### H.49.1. AltOS

AltOS fixes:

- TeleMetrum v1.0 boards use the AT45DB081D flash memory part to store flight data, which is different from later TeleMetrum boards. The AltOS v1.1 driver for this chip couldn't erase memory, leaving it impossible to delete flight data or update configuration values. This bug doesn't affect newer TeleMetrum boards, and it doesn't affect the safety of rockets flying version 1.1 firmware.

### H.49.2. AltosUI

AltosUI new features:

- The "Descent" tab displays the range to the rocket, which is a combination of the over-the-ground distance to the rockets current latitude/longitude and the height of the rocket. As such, it's useful for knowing how far away the rocket is, but difficult to use when estimating where the rocket might eventually land. A new "Ground Distance" field has been added which displays the distance to a spot right underneath the rocket.

AltosUI fixes:

- Creating a Google Earth file (KML) from on-board flight data (EEPROM) would generate an empty file. The code responsible for reading the EEPROM file wasn't ever setting the GPS valid bits, and so the KML export code thought there was no GPS data in the file.
- The "Landed" tab was displaying all values in metric units, even when AltosUI was configured to display imperial units. Somehow I just missed this tab when doing the units stuff.
- Sensor data wasn't being displayed for TeleMini flight computers in Monitor Idle mode, including things like battery voltage. The code that picked which kinds of data to fetch from the flight computer was missing a check for TeleMini when deciding whether to fetch the analog sensor data.

## H.50. Release Notes for Version 1.1

Version 1.1 is a minor release. It provides a few new features in AltosUI and the AltOS firmware and fixes bugs.

### H.50.1. AltOS

AltOS Firmware New Features:

- Add apogee-lockout value. Overrides the apogee detection logic to prevent incorrect apogee charge firing.
- Force the radio frequency to 434.550MHz when the debug clock pin is connected to ground at boot time. This provides a way to talk to a TeleMini which is configured to some unknown frequency.
- Provide RSSI values for Monitor Idle mode. This makes it easy to check radio range without needing to go to flight mode.

AltOS Fixes:

- Fix a bug where the data reported in telemetry packets was from 320ms ago.
- Fix a bug which caused the old received telemetry packets to be retransmitted over the USB link when the radio was turned off and back on.

### H.50.2. AltosUI

AltosUI New Features:

- Make the look-n-feel configurable, providing a choice from the available options.
- Add an 'Age' element to mark how long since a telemetry packet has been received. Useful to quickly gauge whether communications with the rocket are still active.
- Add 'Configure Ground Station' dialog to set the radio frequency used by a particular TeleDongle without having to go through the flight monitor UI.
- Add configuration for the new apogee-lockout value. A menu provides a list of reasonable values, or the value can be set by hand.
- Add Imperial units mode to present data in feet instead of meters.

AltosUI Fixes:

- Fix a bug that caused GPS ready to happen too quickly. The software was using every telemetry packet to signal new GPS data, which caused GPS ready to be signalled after 10 packets instead of 10 GPS updates.
- Fix Google Earth data export to work with recent versions. The google earth file loading code got a lot pickier, requiring some minor white space changes in the export code.
- Changed how flight data are downloaded. Now there's an initial dialog asking which flights to download, and after that finishes, a second dialog comes up asking which flights to delete.

- Re-compute time spent in each state for the flight graph; this figures out the actual boost and landing times instead of using the conservative values provide by the flight electronics. This improves the accuracy of the boost acceleration and main descent rate computations.
- Make AltosUI run on Mac OS Lion. The default Java heap space was dramatically reduced for this release causing much of the UI to fail randomly. This most often affected the satellite mapping download and displays.
- Change how data are displayed in the 'table' tab of the flight monitoring window. This eliminates entries duplicated from the header and adds both current altitude and pad altitude, which are useful in 'Monitor Idle' mode.

## H.51. Release Notes for Version 1.0.1

Version 1.0.1 is a major release, adding support for the TeleMini device and lots of new AltosUI features

### H.51.1. AltOS

#### AltOS New Features

- Add TeleMini v1.0 support.
- Support operation of TeleMetrum with the antenna pointing aft. Previous firmware versions required the antenna to be pointing upwards, now there is a configuration option allowing the antenna to point aft, to aid installation in some airframes.
- Ability to disable telemetry. For airframes where an antenna just isn't possible, or where radio transmissions might cause trouble with other electronics, there's a configuration option to disable all telemetry. Note that the board will still enable the radio link in idle mode.
- Arbitrary frequency selection. The radios in Altus Metrum devices can be programmed to a wide range of frequencies, so instead of limiting devices to 10 pre-selected 'channels', the new firmware allows the user to choose any frequency in the 70cm band. Note that the RF matching circuit on the boards is tuned for around 435MHz, so frequencies far from that may reduce the available range.

#### AltOS Fixes

- Change telemetry to be encoded in multiple 32-byte packets. This enables support for TeleMini and other devices without requiring further updates to the TeleDongle firmware.
- Kalman-filter based flight-tracking. The model based sensor fusion approach of a Kalman filter means that AltOS now computes apogee much more accurately than before, generally within a fraction of a second. In addition, this approach allows the baro-only TeleMini device to correctly identify Mach transitions, avoiding the error-prone selection of a Mach delay.

### H.51.2. AltosUI Application

#### AltosUI New Features

- Add main/apogee voltage graphs to the data plot. This provides a visual indication if the igniters fail before being fired.
- Scan for altimeter devices by watching the defined telemetry frequencies. This avoids the problem of remembering what frequency a device was configured to use, which is especially important with TeleMini which does not include a USB connection.
- Monitor altimeter state in "Idle" mode. This provides much of the information presented in the "Pad" dialog from the Monitor Flight command, monitoring the igniters, battery and GPS status without requiring the flight computer to be armed and ready for flight.
- Pre-load map images from home. For those launch sites which don't provide free Wi-Fi, this allows you to download the necessary satellite images given the location of the launch site. A list of known launch sites is maintained at [altusmetrum.org](http://altusmetrum.org) which AltosUI downloads to

populate a menu; if you've got a launch site not on that list, please send the name of it, latitude and longitude along with a link to the web site of the controlling club to the altusmetrum mailing list.

- Flight statistics are now displayed in the Graph data window. These include max height/speed/accel, average descent rates and a few other bits of information. The Graph Data window can now be reached from the 'Landed' tab in the Monitor Flight window so you can immediately see the results of a flight.

### AltosUI Changes

- Wait for altimeter when using packet mode. Instead of quickly timing out when trying to initialize a packet mode configuration connection, AltosUI now waits indefinitely for the remote device to appear, providing a cancel button should the user get bored. This is necessary as the TeleMini can only be placed in "Idle" mode if AltosUI is polling it.

## H.52. Release Notes for Version 0.9.2

Version 0.9.2 is an AltosUI bug-fix release, with no firmware changes.

### H.52.1. AltosUI

AltosUI fixes:

- Fix plotting problems due to missing file in the Mac OS install image.
- Always read whole eeprom blocks, mark empty records invalid, display parsing errors to user.
- Add software version to Configure AltosUI dialog

## H.53. Release Notes for Version 0.9

Version 0.9 adds a few new firmware features and accompanying AltosUI changes, along with new hardware support.

### H.53.1. AltOS

- Support for TeleMetrum v1.1 hardware. Sources for the flash memory part used in v1.0 dried up, so v1.1 uses a different part which required a new driver and support for explicit flight log erasing.
- Multiple flight log support. This stores more than one flight log in the on-board flash memory. It also requires the user to explicitly erase flights so that you won't lose flight logs just because you fly the same board twice in one day.
- Telemetry support for devices with serial number  $\geq 256$ . Previous versions used a telemetry packet format that provided only 8 bits for the device serial number. This change requires that both ends of the telemetry link be running the 0.9 firmware or they will not communicate.

### H.53.2. AltosUI Application

- Support for telemetry format changes.
- Support for multiple flight logs.

## H.54. Release Notes for Version 0.8

Version 0.8 offers a major upgrade in the AltosUI interface.

### H.54.1. AltosUI Application:

- Post-flight graphing tool. This lets you explore the behaviour of your rocket after flight with a scroll-able and zoom-able chart showing the altitude, speed and acceleration of the airframe along with events recorded by the flight computer. You can export graphs to PNG files, or print them directly.
- Real-time moving map which overlays the in-progress flight on satellite imagery fetched from Google Maps. This lets you see in pictures where your rocket has landed, allowing you to plan recovery activities more accurately.
- Wireless recovery system testing. Prep your rocket for flight and test fire the deployment charges to make sure things work as expected. All without threading wires through holes in your airframe.
- Optimized flight status displays. Each flight state now has it's own custom 'tab' in the flight monitoring window so you can focus on the most important details. Pre-flight, the system shows a set of red/green status indicators for battery voltage, apogee/main igniter continuity and GPS reception. Wait until they're all green and your rocket is ready for flight. There are also tabs for ascent, descent and landing along with the original tabular view of the data.
- Monitor multiple flights simultaneously. If you have more than one TeleDongle, you can monitor a flight with each one on the same computer.
- Automatic flight monitoring at startup. Plug TeleDongle into the machine before starting AltosUI and it will automatically connect to it and prepare to monitor a flight.
- Exports Google Earth flight tracks. Using the Keyhole Markup Language (.kml) file format, this provides a 3D view of your rocket flight through the Google Earth program.

## H.55. Release Notes for Version 0.7.1

Version 0.7.1 is the first release containing our new cross-platform Java-based user interface.

### H.55.1. AltosUI Application

- Receive and log telemetry from a connected TeleDongle device. All data received is saved to log files named with the current date and the connected rocket serial and flight numbers. There is no mode in which telemetry data will not be saved.
- Download logged data from TeleMetrum devices, either through a direct USB connection or over the air through a TeleDongle device.
- Configure a TeleMetrum device, setting the radio channel, callsign, apogee delay and main deploy height. This can be done through either a USB connection or over a radio link via a TeleDongle device.
- Replay a flight in real-time. This takes a saved telemetry log or eeprom download and replays it through the user interface so you can relive your favorite rocket flights.
- Reprogram Altus Metrum devices. Using an Altus Metrum device connected via USB, another Altus Metrum device can be reprogrammed using the supplied programming cable between the two devices.
- Export Flight data to a comma-separated-values file. This takes either telemetry or on-board flight data and generates data suitable for use in external applications. All data is exported using standard units so that no device-specific knowledge is needed to handle the data.
- Speak to you during the flight. Instead of spending the flight hunched over your laptop looking at the screen, enjoy the view while the computer tells you what's going on up there. During ascent, you hear the current flight state and altitude information. During descent, you get azimuth, elevation and range information to try and help you find your rocket in the air. Once on the ground, the direction and distance are reported.