

TeleMetrum

Owner's Manual for the TeleMetrum System

Bdale Garbee

Keith Packard

TeleMetrum: Owner's Manual for the TeleMetrum System

Bdale Garbee

Keith Packard

Copyright © 2010 Bdale Garbee and Keith Packard

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

Table of Contents

1. Introduction and Overview	1
2. Getting Started	2
FAQ	4
3. Specifications	5
4. Handling Precautions	6
5. Hardware Overview	7
6. Operation	8
Firmware Modes	8
GPS	8
Ground Testing	9
Radio Link	9
Configurable Parameters	9
Radio Channel	9
Apogee Delay	10
Main Deployment Altitude	10
Calibration	10
Radio Frequency	10
Accelerometer	10
7. Using Altus Metrum Products	12
Being Legal	12
In the Rocket	12
On the Ground	12
Data Analysis	13
Future Plans	13
How GPS Works	13

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!

The focal point of our community is 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.

Complementing TeleMetrum is TeleDongle, a USB to RF interface for communicating with TeleMetrum. 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.

Chapter 2. Getting Started

This chapter began as "The Mere-Mortals Quick Start/Usage Guide to the Altus Metrum Starter Kit" by Bob Finch, W9YA, NAR 12965, TRA 12350, w9ya@amsat.org. Bob was one of our first customers for a production TeleMetrum, and the enthusiasm that led to his contribution of this section is immensely gratifying and highly appreciated!

The first thing to do after you check the inventory of parts in your "starter kit" is to charge the battery by plugging it into the corresponding socket of the TeleMetrum and then using the USB A to B cable to plug the Telemetrum into your computer's USB socket. The TeleMetrum circuitry will charge the battery whenever it is plugged into the usb socket. The TeleMetrum's on-off switch does NOT control the charging circuitry. When the GPS chip is initially searching for satellites, the unit will pull more current than it can pull from the usb port, so the battery must be plugged in order to get a good 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.

The other active device in the starter kit is the half-duplex TeleDongle rf link. If you plug it in to your computer it should "just work", showing up as a serial port device. If you are using Linux and are having problems, try moving to a fresher kernel (2.6.33 or newer), as there were some ugly USB serial driver bugs in earlier versions.

Next you should obtain and install the AltOS utilities. The first generation software was written for Linux only. New software is coming soon that will also run on Windows and Mac. For now, we'll concentrate on Linux. If you are using Debian, an 'altos' package already exists, see <http://altusmetrum.org/AltOS> for details on how to install it. User-contributed directions for building packages on ArchLinux may be found in the contrib/arch-linux directory as PKGBUILD files. Between the debian/rules file and the PKGBUILD files in contrib, you should find enough information to learn how to build the software for any other version of Linux.

When you have successfully installed the software suite (either from compiled source code or as the pre-built Debian package) you will have 10 executable programs all of which have names beginning with 'ao-'. ('ao-view' is the lone GUI-based program. The rest are command-line based.) You will also have 10 man pages, that give you basic info on each program. And you will also get this documentation in two file types, telemetrum.pdf and telemetrum.html. Finally you will have a couple of control files that allow the ao-view GUI-based program to appear in your menu of programs (under the 'Internet' category).

Both Telemetrum and TeleDongle can be directly communicated with using USB ports. The first thing you should try after getting both units plugged into to your computer's usb port(s) is to run 'ao-list' from a terminal-window (I use konsole for this,) to see what port-device-name each device has been assigned by the operating system. You will need this information to access the devices via their respective on-board firmware and data using other command line programs in the AltOS software suite.

To access the device's firmware for configuration you need a terminal program such as you would use to talk to a modem. The software authors prefer using the program 'cu' which comes from the UUCP package on most Unix-like systems such as Linux. An example command line for cu might be 'cu -l /dev/ttyACM0', substituting the correct number indicated from running the ao-list program. Another reasonable terminal program for Linux is 'cutecom'. The default 'escape' character used by CU (i.e. the character you use to issue commands to cu itself instead of sending the command as input to the connected device) is a '~'. You will need this for use in only two different ways during normal operations. First is to exit the program by sending a '~' which is called a 'escape-disconnect' and allows you to close-out from 'cu'. The second use will be outlined later.

Both TeleMetrum and TeleDongle share the concept of a two level command set in their firmware. The first layer has several single letter commands. Once you are using 'cu' (or 'cutecom') sending (typing) a '?' returns a full list of these commands. The second level are configuration sub-commands accessed using the 'c' command, for instance typing 'c?' will give you this second level of commands (all of which require the letter 'c' to access). Please note that most configuration options are stored only in DataFlash memory, and only TeleMetrum has this memory to save the various values entered like the channel number and your callsign when powered off. TeleDongle requires that you set these each time you plug it in, which ao-view can help with.

Try setting these config ('c' or second level menu) values. A good place to start is by setting your call sign. By default, the boards use 'NOCALL' which is cute, but not exactly legal! Spend a few minutes getting comfortable with the units, their firmware, 'cu' (and possibly 'cutecom') For instance, try to send (type) a 'cr2' and verify the channel change by sending a 'cs'. Verify you can connect and disconnect from the units while in 'cu' by sending the escape-disconnect mentioned above.

Note that the 'reboot' command, which is very useful on TeleMetrum, will likely just cause problems with the dongle. The *correct* way to reset the dongle is just to unplug and re-plug it.

A fun thing to do at the launch site and something you can do while learning how to use these units is to play with the rf-link access of the TeleMetrum from the TeleDongle. Be aware that you *must* create some physical separation between the devices, otherwise the link will not function due to signal overload in the receivers in each device.

Now might be a good time to take a break and read the rest of this manual, particularly about the two "modes" that the TeleMetrum can be placed in and how the position of the TeleMetrum when booting up will determine whether the unit is in "pad" or "idle" mode.

You can access a TeleMetrum in idle mode from the Teledongle's USB connection using the rf link by issuing a 'p' command to the TeleDongle. Practice connecting and disconnecting ('~~' while using 'cu') from the TeleMetrum. If you cannot escape out of the "p" command, (by using a '~~' when in CU) then it is likely that your kernel has issues. Try a newer version.

Using this rf link allows you to configure the TeleMetrum, test fire e-matches and igniters from the flight line, check pyro-match continuity and so forth. You can leave the unit turned on while it is in 'idle mode' and then place the rocket vertically on the launch pad, walk away and then issue a reboot command. The TeleMetrum will reboot and start sending data having changed to the "pad" mode. If the TeleDongle is not receiving this data, you can disconnect 'cu' from the Teledongle using the procedures mentioned above and THEN connect to the TeleDongle from inside 'ao-view'. If this doesn't work, disconnect from the TeleDongle, unplug it, and try again after plugging it back in.

Eventually the GPS will find enough satellites, lock in on them, and 'ao-view' will both auditorially announce and visually indicate that GPS is ready. Now you can launch knowing that you have a good data path and good satellite lock for flight data and recovery. Remember you MUST tell ao-view to connect to the TeleDongle explicitly in order for ao-view to be able to receive data.

Both RDF (radio direction finding) tones from the TeleMetrum and GPS trekking data are available and together are very useful in locating the rocket once it has landed. (The last good GPS data received before touch-down will be on the data screen of 'ao-view'.)

Once you have recovered the rocket you can download the eeprom contents using either 'ao-dumplog' (or possibly 'ao-eeprom'), over either a USB cable or over the radio link using TeleDongle. And by following the man page for 'ao-postflight' you can create various data output reports, graphs, and even kml data to see the flight trajectory in google-earth. (Moving the viewing angle making sure to connect the yellow lines while in google-earth is the proper technique.)

As for ao-view.... some things are in the menu but don't do anything very useful. The developers have stopped working on ao-view to focus on a new, cross-platform ground station program. Mostly you just

use the Log and Device menus. It has a wonderful display of the incoming flight data and I am sure you will enjoy what it has to say to you once you enable the voice output!

FAQ

The altimeter (TeleMetrum) seems to shut off when disconnected from the computer. Make sure the battery is adequately charged. Remember the unit will pull more power than the USB port can deliver before the GPS enters "locked" mode. The battery charges best when TeleMetrum is turned off.

It's impossible to stop the TeleDongle when it's in "p" mode, I have to unplug the USB cable? Make sure you have tried to "escape out" of this mode. If this doesn't work the reboot procedure for the TeleDongle **is** to simply unplug it. 'cu' however will retain it's outgoing buffer IF your "escape out" ('~~') does not work. At this point using either 'ao-view' (or possibly 'cutemon') instead of 'cu' will 'clear' the issue and allow renewed communication.

The amber LED (on the TeleMetrum/altimeter) lights up when both battery and USB are connected. Does this mean it's charging? Yes, the yellow LED indicates the charging at the 'regular' rate. If the led is out but the unit is still plugged into a USB port, then the battery is being charged at a 'trickle' rate.

There are no "dit-dah-dah-dit" sound like the manual mentions? That's the "pad" mode. Weak batteries might be the problem. It is also possible that the unit is horizontal and the output is instead a "dit-dit" meaning 'idle'.

It's unclear how to use 'ao-view' and other programs when 'cu' is running. You cannot have more than one program connected to the TeleDongle at one time without apparent data loss as the incoming data will not make it to both programs intact. Disconnect whatever programs aren't currently being used.

How do I save flight data? Live telemetry is written to file(s) whenever 'ao-view' is connected to the TeleDongle. The file area defaults to ~/altos but is easily changed using the menus in 'ao-view'. The files that are written end in '.telem'. The after-flight data-dumped files will end in .eeprom and represent continuous data unlike the rf-linked .telem files that are subject to the turnarounds/data-packaging time slots in the half-duplex rf data path. See the above instructions on what and how to save the eeprom stored data after physically retrieving your TeleMetrum.

Chapter 3. Specifications

- Recording altimeter for model rocketry.
- Supports dual deployment (can fire 2 ejection charges).
- 70cm ham-band transceiver for telemetry downlink.
- Barometric pressure sensor good to 45k feet MSL.
- 1-axis high-g accelerometer for motor characterization, capable of +/- 50g using default part.
- On-board, integrated GPS receiver with 5hz update rate capability.
- On-board 1 megabyte non-volatile memory for flight data storage.
- USB interface for battery charging, configuration, and data recovery.
- Fully integrated support for LiPo rechargeable batteries.
- Uses LiPo to fire e-matches, support for optional separate pyro battery if needed.
- 2.75 x 1 inch board designed to fit inside 29mm airframe coupler tube.

Chapter 4. Handling Precautions

TeleMetrum is a sophisticated electronic device. When handled gently and properly installed in an airframe, it will deliver impressive results. However, like all electronic devices, there are some precautions you must take.

The Lithium Polymer rechargeable batteries used with TeleMetrum 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 airframe. We often wrap them in suitable scraps of closed-cell packing foam before strapping them down, for example.

The TeleMetrum barometric sensor is sensitive to sunlight. In normal mounting situations, it 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, though, when designing an installation, for example, in a 29mm airframe's see-through plastic payload bay.

The TeleMetrum 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, but also by having a suitable static vent to outside air.

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

Chapter 5. Hardware Overview

TeleMetrum is a 1 inch by 2.75 inch circuit board. It was designed to fit inside coupler for 29mm airframe tubing, but using it in a tube that small in diameter may require some creativity in mounting and wiring to succeed! The default 1/4 wave UHF wire antenna attached to the center of the nose-cone end of the board 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. Given all this, an ideal "simple" avionics bay for TeleMetrum should have at least 10 inches of interior length.

A typical TeleMetrum installation using the on-board GPS antenna and default wire UHF antenna involves attaching only a suitable Lithium Polymer battery, a single pole switch for power on/off, and two pairs of wires connecting e-matches for the apogee and main ejection charges.

By default, we use the unregulated output of the LiPo 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, you can do so by adding a second 2mm connector to position B2 on the board and cutting the thick pcb trace connecting the LiPo battery to the pyro circuit between the two silk screen marks on the surface mount side of the board shown here [insert photo]

We offer two choices of pyro and power switch connector, or you can choose neither and solder wires directly to the board. All three choices are reasonable depending on the constraints of your airframe. Our favorite option when there is sufficient room above the board is to use the Tyco pin header with polarization and locking. If you choose this option, you crimp individual wires for the power switch and e-matches into a mating connector, and installing and removing the TeleMetrum board from an airframe is as easy as plugging or unplugging two connectors. If the airframe will not support this much height or if you want to be able to directly attach e-match leads to the board, we offer a screw terminal block. This is very similar to what most other altimeter vendors provide and so may be the most familiar option. You'll need a very small straight blade screwdriver to connect and disconnect the board in this case, such as you might find in a jeweler's screwdriver set. Finally, you can forego both options and solder wires directly to the board, which may be the best choice for minimum diameter and/or minimum mass designs.

For most airframes, the integrated GPS antenna and wire UHF antenna are a great combination. However, if you are installing in a carbon-fiber electronics bay which is opaque to RF signals, you may need to use off-board external antennas instead. In this case, you can order TeleMetrum with an SMA connector for the UHF antenna connection, and you can unplug the integrated GPS antenna and select an appropriate off-board GPS antenna with cable terminating in a U.FL connector.

Chapter 6. Operation

Firmware Modes

The AltOS firmware build for TeleMetrum has two fundamental modes, "idle" and "flight". Which of these modes the firmware operates in is determined 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 TeleMetrum 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.

At power on, you will hear three beeps ("S" in Morse code for startup) and then a pause while TeleMetrum completes initialization and self tests, and decides which mode to enter next.

In flight mode, TeleMetrum turns on the GPS system, engages the flight state machine, goes into transmit-only mode on the RF link sending telemetry, and waits for launch to be detected. Flight mode is indicated by an audible "di-dah-dah-dit" ("P" for pad) on the beeper, followed by beeps indicating the state of the pyrotechnic igniter continuity. One beep indicates apogee continuity, two beeps indicate main continuity, three beeps indicate both apogee and main continuity, and one longer "brap" sound indicates no continuity. For a dual deploy flight, make sure you're getting three beeps before launching! For apogee-only or motor eject flights, do what makes sense.

In idle mode, you will hear an audible "di-dit" ("I" for idle), and the normal flight state machine is disengaged, thus no ejection charges will fire. TeleMetrum also listens on the RF link when in idle mode for packet mode requests sent from TeleDongle. Commands can be issued to a TeleMetrum in idle mode over either USB or the RF link equivalently. Idle mode is useful for configuring TeleMetrum, for extracting data from the on-board storage chip after flight, and for ground testing pyro charges.

One "neat trick" of particular value when TeleMetrum is used with very large airframes, 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 airframe to launch position, use a TeleDongle to open a packet connection, and issue a 'reset' command which will cause TeleMetrum to reboot, realize it's now nose-up, and thus choose 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!

GPS

TeleMetrum includes a complete GPS receiver. See a later section for a brief explanation of how GPS works that will help you understand the information in the telemetry stream. The bottom line is that the TeleMetrum GPS receiver needs to lock onto at least four satellites to obtain a solid 3 dimensional position fix and know what time it is!

TeleMetrum provides backup power to the GPS chip any time a LiPo battery is connected. This allows the receiver to "warm start" on the launch rail much faster than if every power-on were a "cold start" for the GPS receiver. In typical operations, powering up TeleMetrum on the flight line in idle mode while performing final airframe 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.

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 RF link central to the Altus Metrum system, this can be accomplished in a TeleMetrum-equipped rocket without as much work as you may be accustomed to with other systems. It can even be fun!

Just prep the rocket for flight, then power up TeleMetrum while the airframe is horizontal. 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. Then, establish an RF packet connection from a TeleDongle-equipped computer using the P command from a safe distance. You can now command TeleMetrum to fire the apogee or main charges to complete your testing.

In order to reduce the chance of accidental firing of pyrotechnic charges, the command to fire a charge is intentionally somewhat difficult to type, and the built-in help is slightly cryptic to prevent accidental echoing of characters from the help text back at the board from firing a charge. The command to fire the apogee drogue charge is 'i DoIt drogue' and the command to fire the main charge is 'i DoIt main'.

Radio Link

The chip our boards are based on incorporates 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, TeleMetrum firmware listens for an RF connection when it's in "idle mode" (turned on while the rocket is horizontal), which allows us to use the RF link to configure the rocket, do things like ejection tests, and extract data after a flight without having to crack open the airframe. However, when the board is in "flight mode" (turned on when the rocket is vertical) the TeleMetrum 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 and out over the RF link in case the rocket crashes and we aren't able to extract data later...

We don't use a 'normal packet radio' mode because they're just too inefficient. The GFSK modulation we use is just FSK with the baseband pulses passed through a Gaussian filter before they go into the modulator to limit the transmitted bandwidth. When combined with the hardware forward error correction support in the cc1111 chip, this allows us to have a very robust 38.4 kilobit data link with only 10 milliwatts of transmit power, a whip antenna in the rocket, and a hand-held Yagi on the ground. We've had a test flight above 12k AGL with good reception, and calculations suggest we should be good to 40k AGL or more with a 5-element yagi on the ground. We hope to fly boards to higher altitudes soon, and would of course appreciate customer feedback on performance in higher altitude flights!

Configurable Parameters

Configuring a TeleMetrum board for flight is very simple. Because we have both acceleration and pressure sensors, there is no need to set a "mach delay", for example. The few configurable parameters can all be set using a simple terminal program over the USB port or RF link via TeleDongle.

Radio Channel

Our firmware supports 10 channels. The default channel 0 corresponds to a center frequency of 434.550 Mhz, and channels are spaced every 100 khz. Thus, channel 1 is 434.650 Mhz, and channel 9 is 435.550 Mhz. At any given launch, we highly recommend coordinating who will use each channel and when to avoid interference. And of course, both TeleMetrum and TeleDongle must be configured to the same channel to successfully communicate with each other.

To set the radio channel, use the 'c r' command, like 'c r 3' to set channel 3. As with all 'c' sub-commands, follow this with a 'c w' to write the change to the parameter block in the on-board DataFlash chip.

Apogee Delay

Apogee delay is the number of seconds after TeleMetrum detects flight apogee that the drogue charge should be fired. In most cases, this should be left at the default of 0. However, if you are flying redundant electronics such as for an L3 certification, you may wish to set one of your altimeters to a positive delay so that both primary and backup pyrotechnic charges do not fire simultaneously.

To set the apogee delay, use the [FIXME] command. As with all 'c' sub-commands, follow this with a 'c w' to write the change to the parameter block in the on-board DataFlash chip.

Main Deployment Altitude

By default, TeleMetrum will fire the main deployment charge at an elevation of 250 meters (about 820 feet) above ground. We think this is a good elevation for most airframes, but feel free to change this to suit. In particular, if you are flying two altimeters, you may wish to set the deployment elevation for the backup altimeter to be something lower than the primary so that both pyrotechnic charges don't fire simultaneously.

To set the main deployment altitude, use the [FIXME] command. As with all 'c' sub-commands, follow this with a 'c w' to write the change to the parameter block in the on-board DataFlash chip.

Calibration

There are only two calibrations required for a TeleMetrum board, and only one for TeleDongle.

Radio Frequency

The radio frequency is synthesized from a clock based on the 48 Mhz crystal on the board. The actual frequency of this oscillator must be measured to generate a calibration constant. While our GFSK modulation bandwidth is wide enough to allow boards to communicate even when their oscillators are not on exactly the same frequency, performance is best when they are closely matched. Radio frequency calibration requires a calibrated frequency counter. Fortunately, once set, the variation in frequency due to aging and temperature changes is small enough that re-calibration by customers should generally not be required.

To calibrate the radio frequency, connect the UHF antenna port to a frequency counter, set the board to channel 0, and use the 'C' command to generate a CW carrier. Wait for the transmitter temperature to stabilize and the frequency to settle down. Then, divide 434.550 Mhz by the measured frequency and multiply by the current radio cal value show in the 'c s' command. For an unprogrammed board, the default value is 1186611. Take the resulting integer and program it using the 'c f' command. Testing with the 'C' command again should show a carrier within a few tens of Hertz of the intended frequency. As with all 'c' sub-commands, follow this with a 'c w' to write the change to the parameter block in the on-board DataFlash chip.

Accelerometer

The accelerometer we use has its own 5 volt power supply and the output must be passed through a resistive voltage divider to match the input of our 3.3 volt ADC. This means that unlike the barometric sensor, the output of the acceleration sensor is not ratiometric to the ADC converter, and calibration is required. We also support the use of any of several accelerometers from a Freescale family that includes at least +/- 40g,

50g, 100g, and 200g parts. Using gravity, a simple 2-point calibration yields acceptable results capturing both the different sensitivities and ranges of the different accelerometer parts and any variation in power supply voltages or resistor values in the divider network.

To calibrate the acceleration sensor, use the 'c a 0' command. You will be prompted to orient the board vertically with the UHF antenna up and press a key, then to orient the board vertically with the UHF antenna down and press a key. As with all 'c' sub-commands, follow this with a 'c w' to write the change to the parameter block in the on-board DataFlash chip.

The +1g and -1g calibration points are included in each telemetry frame and are part of the header extracted by ao-dumplog after flight. Note that we always store and return raw ADC samples for each sensor... nothing is permanently "lost" or "damaged" if the calibration is poor.

Chapter 7. Using Altus Metrum Products

Being Legal

First off, in the US, you need an [amateur radio license](../Radio) or other authorization to legally operate the radio transmitters that are part of our products.

In the Rocket

In the rocket itself, you just need a [TeleMetrum](../TeleMetrum) board and a LiPo rechargeable battery. An 860mAh battery weighs less than a 9V alkaline battery, and will run a [TeleMetrum](../TeleMetrum) for hours.

By default, we ship TeleMetrum with a simple wire antenna. If your electronics bay or the airframe it resides within is made of carbon fiber, which is opaque to RF signals, you may choose to have an SMA connector installed so that you can run a coaxial cable to an antenna mounted elsewhere in the rocket.

On the Ground

To receive the data stream from the rocket, you need an antenna and short feedline connected to one of our [TeleDongle](../TeleDongle) units. 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.

Right now, all of our application software is written for Linux. However, because we understand that many people run Windows or MacOS, we are working on a new ground station program written in Java that should work on all operating systems.

After the flight, you can use the RF link to extract the more detailed data logged in the rocket, or you can use a mini USB cable to plug into the TeleMetrum board directly. Pulling out the data without having to open up the rocket is pretty cool! A USB cable is also how you charge the LiPo 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 waypoint for the last reported rocket position before touch-down. This makes looking for your rocket a lot like Geo-Caching... just go to the waypoint and look around starting from there.

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 and Bdale both currently own and use the Yaesu VX-7R at launches.

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

1. an antenna and feedline
2. a TeleDongle

3. a notebook computer
4. optionally, a handheld 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. [<http://www.arrowantennas.com/>] The 440-3 and 440-5 are both good choices for finding a TeleMetrum-equipped rocket when used with a suitable 70cm HT.

Data Analysis

Our software makes it easy to log the data from each flight, both the telemetry received over the RF link during the flight itself, and the more complete data log recorded in the DataFlash memory on the TeleMetrum board. Once this data is on your computer, our postflight 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 data file useable 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.

Future Plans

In the future, we intend to offer "companion boards" for the rocket that will plug in to TeleMetrum to collect additional data, provide more pyro channels, and so forth. A reference design for a companion board will be documented soon, and will be compatible with open source Arduino programming tools.

We are also working on the design of a hand-held ground terminal that will allow monitoring the rocket's status, collecting data during flight, and logging data after flight without the need for a notebook computer on the flight line. Particularly since it is so difficult to read most notebook screens in direct sunlight, we think this will be a great thing to have.

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...

How GPS Works

Placeholder.