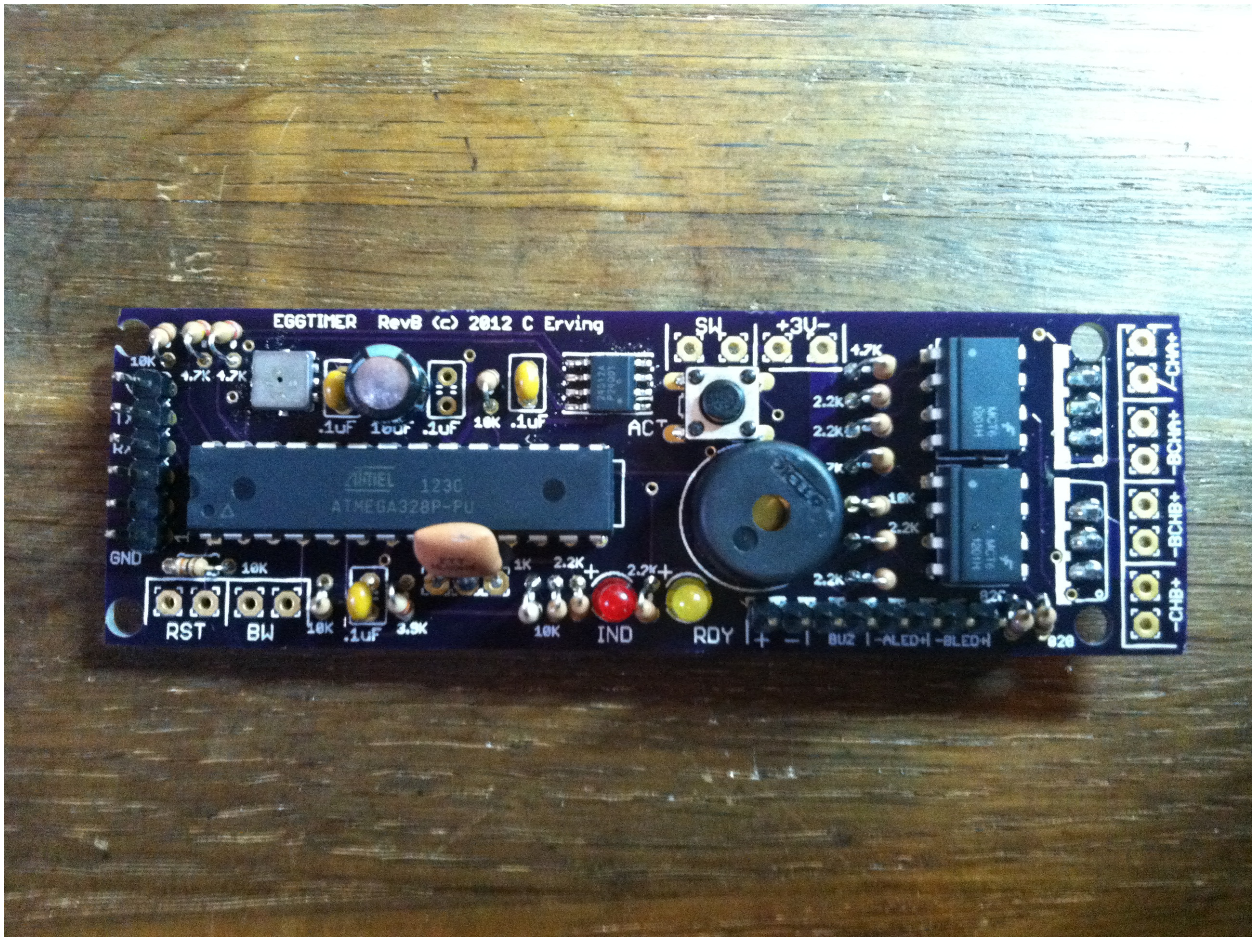
Eggtimer Flight Computer
User's Manual

Release 1.44 Board RevC



Eggtimer Features

Altimeter rated to 29,500 ft ASL, resolution to approx 1 ft.

Logs data for last 32 flights, “beeps” out maximum altitude of last flight

Flight data can be downloaded to a computer using USB interface cable (included) to .CSV format, for use with virtually any spreadsheet, graphing, or statistical analysis software package

Reports Elapsed Flight Time, Altitude @ Time, Raw Velocity @ Time, and Kalman-filtered Altitude and Velocity @ Time

Flight memory is non-volatile, so all settings and flight data are retained with the battery disconnected, and interrupted flights can still produce a data graph

Fully configurable flight parameters including sampling rates, deployment time/altitudes, etc.

Eight pre-programmed flight profiles, so you can select commonly-used flight profiles in the field without needing a computer

Will work with any 3v-4v power source: 3v lithium batteries, two 1.5v alkaline batteries, 3.6v NiMh battery packs, or 3.7v LiPo batteries

Two programmable deployment channels: Main and Drogue/Airstart. Supports two-stage multiple-deployment flights using motor ejection for booster recovery and second-stage drogue deployment

Three logic-level outputs, triggered at nose-over, Main deployment altitude, and landing detect

Supports igniters with a voltage rating of up to 30V, up to 8A, 65W maximum

Deployment outputs also support standard analog PWM hobby servos for non-pyro deployment

Uses optoisolators and bipolar driver transistors for total isolation of computer and ignition power and immunity from false ESD or RF triggering and “brownout” protection

Uses separate battery (not included) for external deployment device; each channel can have a separate battery so servos and igniters can be mixed

Breakwire support for arming Airstarts, 3-factor triggering for safety

Supports remote Power and Reset/Hold switches, and remote Ready/Buzzer indicators

Outputs for external igniter continuity LEDs, current-limited

Programmable using standard VT100/ANSI text terminal programs, compatible with virtually all computers (PC, Mac, Linux)

Firmware can be updated from a standard PC using the serial cable

Weight about 20 grams, size approx. 3.9” x 1”; fits in 29mm engine tube or BT-55 body tube

Disclaimers, Legal Stuff, Etc.

The Eggtimer is meant to be used for hobby and experimental rocketry purposes. Although hobby rocketry has an admirable safety record, largely due to the efforts of the good people at the National Association of Rocketry (NAR) and the Tripoli Rocketry Association (TRA), rocketry can be dangerous if proper safety precautions are not observed. This is particularly true with some of the advanced techniques like pyrotechnic parachute deployment and igniting rocket motors in flight ("Airstarts"). People can and have been seriously injured by not following recognized and accepted safety practices. We cannot be responsible for your actions.

We ***strongly*** recommend that if you are not a member of either the NAR or the TRA, you join one of them, join a local rocketry club, and pick the brains of experienced members before you try any kind of multiple deployment or airstart flight. The safety information included in these instructions is by no means comprehensive or complete, and is no substitute for the supervision and advice of experienced rocketeers.

Limited Warranty

Eggtimer Rocketry warrants that all of the parts on the packing list of the Eggtimer kit have been included, and that they are all in working condition. If you are missing something, contact us immediately at support@EggtimerRocketry.com and we will send you whatever it is that you are missing. If you are missing something really egregious (like the PC board or the processor, for example), we may ask you to return the entire kit unbuilt, we will send you a prepaid shipping label for this purpose. We'd especially like to see the packing list so we can figure out what went wrong so it doesn't happen again…

If your Eggtimer does not work properly after assembly, take a deep breath, get out the magnifying glass and a good light, and see if you have inadvertently created a solder bridge somewhere. Chances are pretty good that you have, or that you have installed a part incorrectly. We are a very small company and we just don't have the resources to repair your board, but we will be more than happy to give you advice and we might be able to help you find your error if you send us some high resolution pictures, to support@EggtimerRocketry.com . We cannot take responsibility for your assembly techniques; if you do not have experience building kits of this nature, we recommend that you enlist some help. (Another reason for joining a rocketry club, there is usually at least one electronically-inclined member who can be bribed with a beverage or two to give you a hand. Engineering types love a challenge, especially it it's easy for them but hard for you.)

Eggtimer Rocketry warrants that when properly assembled the Eggtimer will perform substantially according to the published documentation. This means that we spent a lot of time trying to ensure that it's going to work the way that we say it does, and we try to fix things that don't quite work right in a reasonable time. Nevertheless, we can not and do not warrant that the Eggtimer is perfect and will meet every rocketry purpose, for the simple reason that we can't test every possible rocket/motor/environmental combination. It is the buyer's responsibility to determine the suitability of the Eggtimer for their particular purpose. If you have a problem with this, please contact us and we will be happy to send you a prepaid return label for your unbuilt Eggtimer kit and we will refund your purchase price.

A General Introduction to Flight Computers

A flight computer is used to record the flight path of a rocket, and to optionally deploy recovery devices. Typically, you can use it to find out the maximum altitude (apogee) of your rocket, the maximum velocity of the rocket, the maximum acceleration (G forces) on your rocket, and other flight characteristics. Being able to see these after a flight helps to validate your own designs, and a flight computer is essential for high-power rockets since the motors used in those rockets often don't have built-in ejection charges. Typically, such rockets have two parachutes: a smaller "drogue" parachute that is deployed at apogee, and a larger "main" parachute that is deployed at a much lower altitude. The drogue parachute keeps the rocket from drifting too far away… if the main parachute is deployed at 10,000', it is possible that your rocket could drift over a mile from the launch site. Using two-parachute deployment scheme typically keeps the drift down to 1000' or less.

They can also be used to "airstart" motors in flight, i.e. to ignite the second (or even third) stage motors. Doing this is a more advanced technique than multiple recovery deployment, and it requires accurate software flight simulation in order to select the proper time and/or altitude to ignite the motor. Typically, the second stage is ignited either when burnout of the first stage motor is detected or after a short delay afterwards.

Flight computers can use either a barometric pressure altimeter, an accelerometer "G-sensor", or both. Each one has its own advantages and disadvantages. A barometric pressure sensor can be used to directly determine altitude, and when read at known intervals can be used to calculate velocity between intervals with fairly reasonable accuracy. You can in theory use one to determine acceleration, but slight errors tend to be multiplied resulting in much larger errors. Since they read barometric pressure which depends on temperature, and may be thrown off by aerodynamic issues such as the transition to supersonic speeds, careful consideration needs to be taken to ensure that the altitude readings are accurate. Barometric pressure sensors tend to be relatively slow, so the sampling rate may be limited by the speed of the sensor. Also, errors in barometric pressure readings are inevitable during high-speed flight, because the pressure is still changing as the sensor is being read. Derived data such as velocity may be inaccurate at angles that vary significantly from the vertical, since the recorded altitude will be less than the distance that the rocket actually travels during that time interval.

An accelerometer uses the forces on the rocket to determine when the motor has fired and burned out, and can be used to determine velocity over a known interval with better accuracy than an altimeter. Since it reacts directly to the forces on the rocket, it is very easy to detect reaction-based events such as burnout, and therefore is the best sensor to use for "air starting" a second stage motor in flight. It can also be used to determine altitude by integrating (adding up) the differences in calculated distance for each time interval, however since this is "distance-traveled" reading it may not be accurate at angles much above vertical. Also, high-G accelerometers (> 50 G, which may be experienced by high-power rockets) are expensive chips, costing a lot more than a barometer chip. In addition, they only work well when the rocket is going up, once it slows down before apogee the acceleration is minimal and therefore readings will not be accurate. On the plus side, they tend to be very fast, so higher sampling rates can be used compared to barometric pressure sensors.

Generally, accelerometers are more accurate for events logged when the rocket is going up, and barometric pressure altimeters are more accurate for events that happen when the rocket is going down. However, since most of the critical events such as ejection tend to occur in the descent phase, an altimeter-based flight computer can do almost everything that a computer with both an altimeter and an accelerometer can do, at much less cost. With careful software data filtering and a judicious choice of settings it is possible to detect burnout with relatively decent accuracy with a barometer-based flight computer, and it is possible to use a barometric based flight computer to do Airstarts if at least two non-barometric events are used to trigger it.

Meet the Eggtimer

The Eggtimer is an altimeter-based flight computer with two deployment channels. It can record up to 32 flights, of varying length; typical flight time for most rockets is 1-2 minutes or less, but flights can be recorded for much longer periods if necessary. Altitude samples are taken at programmable intervals, depending on the phase of the flight: Burn, Coast, or Descent. In general, Burn and Coast intervals are the shortest, with the descent interval being much longer because the rocket is descending at a relatively slow and constant rate so a faster rate won't result in more meaningful data. There is a wide variety of settings available to accommodate different rocket and motor combinations.

The Eggtimer is designed primarily to be used with medium power rockets, typically from "D" to "G" power, or Level 1 and 2 HPR rockets using "H" through "K" power. The Main (Channel A) deployment channel is programmable so that it may be triggered at apogee or at a fixed altitude from 100-2000 feet, and you can select the number of seconds that the trigger is turned on. The "Channel B" output can be used as either a Drogue deployment channel that fires at just after apogee, or it can be used to airstart a second stage a programmed time after reaching Launch Detect Altitude. Since most of the motors that the Eggtimer is designed to be used with come with built-in ejection charges, the Main deployment channel can be used for the main parachute, the built-in ejection charge can be used for the drogue parachute, and the Channel B output can be used for airstarts, so the Eggtimer can support two-stage multiple deployment rockets.

The deployment channels are capable of triggering a medium-current Estes igniter or #36 nichrome wire using an appropriate battery (such as a small 7.4v LiPo pack). You can also use a low-current igniter such as a Quest Q2G2 with a single 3.7v LiPo battery. The Eggtimer supports inexpensive pulse-width modulated (PWM) analog servos, which are typically used with radio control models, so it can also be used with non-pyro deployment mechanisms, and can even be used with large water rockets.

There are also three unbuffered logic-level outputs, they are set to turn on at nose-over, main deployment altitude (shadowing the Main channel), and at landing detect. With the addition of suitable buffer and driver circuitry, these channels can provide redundancy for the primary channels, and the AUX3 (landing) channel can be used to turn on a sounding or tracking device when the rocket lands.

Programming is performed using a standard VT100/ANSI serial terminal program, connected to your computer with a USB-serial cable. Since there is no special software required, you can use just about any computer, running Windows, Mac, Linux, or just about anything else you can come up with.   Flights can be downloaded to a standard CSV-formatted file, and can be imported into virtually any spreadsheet or data analysis program to produce flight graphs.

The firmware on the processor chip performs real-time data analysis during flights to detect and report significant flight events such as launch, burnout, apogee, deployment, and landing. Hardware breakwire support is provided to provide an extra level of redundancy for airstarts, this ensures that the rocket has indeed left the pad and that a sudden gust of wind isn't going to trick the Eggtimer into thinking that it's reached the launch detect altitude.

Getting to know your Eggtimer

Although the Eggtimer seems relatively simple, you will want to familiarize yourself with it BEFORE you install it in a rocket, and certainly before you try flying it.

Battery (3V) – Two solder pads, marked "+" and "-", for connecting the Eggtimer battery. Typical draw is about 8 ma when running, recommended battery voltage is 3v-4v.

Push Button (ACT) – Performs various actions when pressed.

Indicator (IND) – Glows solid or blinking, depending on the status of the Eggtimer.

Ready Indicator (RDY) – Glows on when ready for launch, blinks when flight is done or the Eggtimer is in download mode.

Beeper (BUZ) - "Beeps" out status, warnings, altitude, and other important notifications.

Channel A (Main) Deployment (CHA) - Two solder pads, marked "+" and "-" for connecting an electric match or other low-to-medium current device. Deployment delay/altitude is programmable. Maximum 8A current, 30 VDC, 60 W. You can also connect the input for a standard analog hobby servo to the "+" terminal (along with a 39 ohm ¼ W resistor in parallel) for non-pyro deployment.

Channel A (Main) Battery (BCHA) - Two solder pads, marked "+" and "-", for connecting an external battery for the deployment device (if used). Typically, these batteries are 3.7v LiPo cells for low-current igniters such as a Quest G2Q2, or 9-12v alkaline batteries for #36 nichrome wire or other igniters. The Eggtimer is capable of handling up to 30v, so you can ignite even an

Aerotech Copperhead igniter with it, assuming you have a strong enough battery (i.e. 14.4v LiPo battery).

Channel B Deployment (CHB) - Two solder pads, marked "+" and "-" for connecting an electric match or other low-to-medium current device. Deployment delay/altitude is programmable. Maximum 8A current, 30 VDC, 60 W. You can also connect the input for a standard analog hobby servo to the "+" terminal (along with a 39 ohm ¼ W resistor in parallel) for non-pyro deployment.

Channel B Battery (BCHB) – Separate battery terminals for the Channel B deployment. This is provided in case you want to use two different batteries for Main and Channel B; this allows you to mix pyro outputs and servos, since servos generally use 6v or less. If you are using both Channel A (Main) and Channel B, and they are OK on the same battery (i.e. they're both either servos or both igniters), then you can jumper the BCHA and BCHB terminals together: + to +, and – to - .

AUX1 Output – The AUX1 output turns ON (+V) for two seconds, at nose-over detect. It can be used to provide redundancy for the CHB channel in drogue (nose-over) mode, or it can be used to trigger a drogue parachute if CHB is being used in Airstart mode. This is an unbuffered (logic-level) output, a high-current driver circuit will be required to trigger a load of over 20 mA or over +V (typically 3V-4V). See the Eggtimer web site (www.EggtimerRocketry) for details.

AUX2 Output – The AUX2 output "shadows" the function of the CHA (MAIN) deployment output. It can be used to provide redundancy for the CHA output. Note that if CHA is disabled (set to "0"), the AUX2 channel will also be disabled. This is an unbuffered circuit, see the description of AUX1 for details.

AUX3 Output – The AUX3 output turns ON (+V) when a landing has been detected (AGL of less than 30 ft for at least 5 seconds), and stays on until the Eggtimer is reset. This is an unbuffered output, see the description of AUX1 for details.

Data Port – 6-pin FTDI-compatible standard header connector for connecting the USB-serial data cable for programming and downloading data, and also for connecting a telemetry device such as an RF serial transmitter (not included).

External Switch/Indicator Options

The Eggtimer has several connections for external switches and/or indicators.

Power Switch (SW)

An external power switch will make it much easier to use the Eggtimer, because you won't keep having to disconnect the battery, or removing the jumper across the SW terminals. If you are using multiple deployment, we STRONGLY recommend using an external power switch so you can turn off the Eggtimer while your model is undergoing the safety check by the RSO; in fact, most NAR and Tripoli sanctioned clubs require this. (The alternative is to have an external Reset switch/jumper; you must have at least ONE of these, but both would be preferable if you are doing multiple deployment, and may be required by your club if you are airstarting).

Just about any switch will work with the Eggtimer because the voltage is only about 3v and the current is < 10 ma. However, you need to use a switch that can handle the expected G forces that you expect the rocket to experience during flight. This is one more reason to use a flight simulation program such as RocSim or Open Rocket; you need to know what these forces are going to be to pick a switch.

In general, we recommend that if you use a slide switch that it is mounted so that it slides sideways, not up and down. This will prevent G forces from possibly causing the switch to "bounce", interrupting the power to the Eggtimer, which is not a good thing. Any slide switch that you use must be rated to at least twice the G forces that you are likely to see… a $1 Radio Shack special isn't going to cut it, spend a few bucks and get a high-quality switch.

You can also use a "push-on, push-off" type switch. Many users have had good success with them, also mounted laterally. You can put the switch just behind one of the air ports, and actuate it by pushing a small pin/wire through the hole. Just like with slide switches, spend the money to get a good quality switch.

A better option would be a more positive switch, such as a rotary switch or a screw-type switch that locks down positively. Since the forces on rockets are almost entirely along the longitudinal axis of the rocket, the contacts on a rotary switch are unlikely to be interrupted by G forces. A good choice is the Schurter 033.4501 rotary switch, they cost about $5. This is a DPDT rotary switch originally designed to be a 120v/220v selector switch, but it works very well for our purposes. You can get them from a number of online rocketry suppliers, or you can order one directly from Allied Electronics, a direct distributor for Schurter products. They're actually about a buck cheaper from Allied, but you'll have to pay shipping, so chances are pretty good that you're gonna come out ahead if you buy it from one of the rocketry suppliers because you're probably buying a bunch of stuff from them anyway.

Featherweight Rocketry also makes a good cheap screw-type switch, they use a screw to positively lock down the contacts and completely eliminate any possibility of the switch being jarred open. You can also make your own screw switch, Google around and you can probably find some good examples.

Reset Switch (RST)

When shorted, the Reset terminals put the Eggtimer “on hold”, i.e. resets it so that it does nothing. When it is in reset mode, it draws very little power from the battery, < 2 ma. All outputs are held low, so this is the “safe” mode when multiple deployment systems are used; they cannot be triggered while the Eggtimer is reset. This can be used to put the Eggtimer in low power mode while it’s sitting on the pad for a long time, and can be released shortly before launch to help save battery life.

When the reset switch is opened, the Eggtimer begins the startup sequence. You will hear a continuous “beep-beep” sound, the programming screen is displayed for 30 seconds, and the flight sequence begins if no keys are pressed within this time or the button is not held down to select a pre-programmed flight profile. After approximately 20 more seconds, the Eggtimer begins taking altitude readings and recording, and chirps until launch to let you know that it’s ready for flight.

If you use a multiple deployment system, we STRONGLY recommend that you have some sort of external switch or shorting plug to hold the Eggtimer in reset mode until shortly before launch. A lot of people have had good results with 2.5mm earphone jacks, you simply wire each lug to the RST terminals then you short the leads on the plug together. Hang a “Remove Before Flight” ribbon on the end of the plug, and you’re ready to go; simply pull the plug out when you’re ready to launch to start the Eggtimer flight sequence.

You can also use the same rotary switch listed above in the Power Switch section, they’re a little pricey at $5 each but they work very well. For larger rockets where you have the room and are spending a lot of money anyway, it’s definitely the way to go.

Breakwire (BW)

A “breakwire” is used to definitively indicate that the rocket has lifted off. This technique is used on “real” rockets, and works very well for arming multiple deployment rockets as well, particularly for airstarts. Basically, when a rocket lifts off a wire that is connected between the rocket and the pad breaks an electrical connection, and that event triggers a liftoff sequence.

In the case of the Eggtimer, the breakwire is used to arm Channel B in airstart mode. This provides an additional level of safety to prevent accidental airstart ignition; for the igniter to fire, the breakwire must be tripped AND the rocket must have reached the LDA altitude AND the Burn Timer/Airstart Timer must have expired after LDA. This virtually eliminates any possibility of the second stage igniting on the pad. As we have mentioned before, you ***must*** shunt the igniter until just prior to launch.

The downside of breakwires is that it’s one more thing you have to set up on the pad. See Appendix D for airstart and breakwire implementation tips.

Indicator Headers – .1"-spaced headers with outputs for the following external indicators:

Ready LED – In parallel with the on-board Ready LED.
BUZ – In parallel with the on-board buzzer. This must be a pizo element.
ALED – Channel A continuity LED
BLED – Channel B continuity LED

Ready LED (RDY, +/-)

This is an external extension of the on-board RDY (yellow) LED.

The READY LED on the Eggtimer is used to indicate that either the Eggtimer is in flight mode (solid ON), the flight is complete (ON with the IND LED also ON), or is ready for download (flashing ON). It also flashes along with the buzzer to indicate peak altitude after the flight, and flashes with the IND light after programming to let you know that the Eggtimer is halted. You can connect an external LED to the READY terminals so that you can visually confirm that the Eggtimer is ready for flight to begin or has completed. You will need an LED capable of lighting with approximately 10 ma of current or less, most medium-to-high intensity LEDs will work fine. The "long" lead of the LED should be connected to the "+" terminal, the shorter one should be connected to the "-" terminal. If you are using an external LED in a holder with leads, the red wire should be connected to the "+" terminal and the black wire should be connected to the "-" terminal.

We recommend using an "ultra bright" LED, they are easier to see outdoors but they will draw more current. The reality is that the amount of time that the Ready LED is going to be on is very small, and any decent battery is going to run a long time without worrying about an extra 10 ma of current draw for a few minutes.

BUZ – Buzzer

This is an extension of the on-board pizo buzzer. The on-board buzzer isn't meant to be very loud, it's designed to be used as an indicator when you're right next to the Eggtimer to tell you that it's ready, or to beep out the altitude after the flight. If you want to add a louder one, you can use these terminals. The buzzer may be a pizo electric element, or it can be a pizo buzzer with a built-in driver. It must draw less than 20 mA.

If you want to add a VERY LOUD buzzer that helps you find your rocket, and you have a deployment channel available, you can use the unused deployment channel in continuous mode with a 9v-12v battery and a Mallory Sonalert® buzzer. They're about $10. If you're not using CHB, just put it in drogue mode (option 1) and change the On-Time to be zero; if you're not using CHA, put it in Main deployment mode (option 1), choose a low deployment altitude, maybe 200' or so, and set the On-Time to zero. You can also use the AUX3 channel with a small relay to turn on the buzzer; you may need a driver circuit, see the Eggtimer web site (www.EggtimerRocketry.com) for details. Similarly you can also use the nice little sounders sold by Adept Rocketry, they make a very distinct sound that's hard to miss even with other background noise present.

ALED/BLED - Deployment Channel LEDs

ALED/BLED are designed to provide a visual indicator of deployment channel power and igniter continuity. This lets you know immediately after connecting the igniter and powering up the channel if something isn't right. They use relatively little current, so leaving them on isn't likely to be a significant drain on your deployment battery.

The ALED/BLED outputs are current-limited with 820 ohm resistors, a 9v battery on the outputs will source about 9ma of current to the LEDs. This is more than enough to brightly light up a high-brightness T-1 sized LED. Note that these outputs are polarized, they are marked +/-, so you need to make sure that you have these connected properly; the long lead of the LEDs goes to the + output. If you're using an LED that's pre-wired in a holder, connect the red wire to the "+" pad and the black wire to the "-" pad.

Eggtimer Battery Options

The Eggtimer requires a battery with a nominal voltage of 3.0v-4.0v. The typical current draw is about 8-10 ma when operating, and 2 ma when is Reset/Hold mode. This means that you have several battery options:

- 3v Lithium batteries, such as a CR123 or CR2
- 3v Lithium "coin cells", such as a CR2032, *if they are used for a very short time*
- 3v or 3.6v rechargeable Lithium batteries
- Two 1.5v alkaline batteries wired in series for 3v, typically size AAA or AA
- 3.6v rechargeable Nickel Metal Hydride (NiMh) battery packs
- 3.7v Lithium Ion (LiPo) cells

Because of the wide variety of rocket sizes and the different types of installations that are possible, the Eggtimer does not come with a battery. There are a wide variety of battery options available, which one you choose will depend on many factors:

1) The size of the rocket – Small rockets may require a smaller battery than a large rocket. If you are trying to put the Eggtimer into the smallest possible space, a 3v Lithium "coin cell" battery may work, but see the notes below.

2) How often you wish to replace the battery – Smaller lower-capacity batteries will have to be replaced (or recharged) more often than larger batteries, which is the trade-off for the smaller size

3) How much room you have for a battery – If you have a 3”, 4 pound I-powered rocket you probably have plenty of room for a 250 maH 3.6v NiMH battery, and don’t care so much about a few extra grams. Not so true if you’re flying it in a BT-55 “D”-powered rocket…

4) How much the battery costs – Rechargeable LiPo batteries are nice, but you may need a special charger for them. NiMH or plain old alkaline batteries are heavier, but they cost a whole lot less to use. Non-rechargeable Lithium cells (i.e. CR2, CR123) work very well, provide a lot of current, are cheap and readily available, and are relatively lightweight compared to alkaline batteries and NiMH packs.

Lithium coin cell batteries (i.e. CR2032) will work with the Eggtimer, however they have a relatively small capacity compared to most of the other battery options. Specifically, a CR2032 battery is rated at 225 maH, but this rating is for an extremely low draw, 80 MICRO-amps, which is what you might get from a very low power device like a digital watch or calculator. The Eggtimer draws significantly more current that this, so you may only get 100 maH of capacity out of the battery, and that will drop even less if the battery gets warm (as in sitting out on the pad in the desert in 110° weather). If you want to use a coin cell battery, use the largest one you can fit in the rocket (CR2450’s are the best since they have the largest capacity), and test them with a Digital Voltmeter while the Eggtimer is powered on BEFORE EVERY FLIGHT.

General Battery Procedures

1) TEST YOUR BATTERY WITH A DVM BEFORE EVERY FLIGHT!!!

If you do not have one, buy an inexpensive digital voltmeter and check the battery voltage while the Eggtimer is powered ON, before EVERY flight. The voltage across the two Battery terminals (B+ and B-) should be between 2.8v and 4.0v. (It may be a little higher if you have just recharged the battery, i.e. LiPo batteries may show up to 4.2v when fully charged; that’s OK, the Eggtimer will be fine with it). If your battery reads under 2.8v, recharge it or dispose of it and use another battery. NEVER fly with a battery under 3.0v if you are using electronic deployment.

Similarly, test the deployment batteries as well if you are using electronic deployment. Check your igniter vendor’s guidelines for recommended voltage, and replace the battery if it’s close to the minimum value, or if the voltage is less than 80% of its rated value, whichever is highest. For example, a Quest Q2G2 igniter will fire with as little as 3.7v (a fully-charged LiPo cell), but if you’re using as 9v battery you should replace it if it’s under 7.2v (80% of its rated value) because its voltage drops rapidly as it begins to run down.

The Eggtimer’s built-in battery tester is designed to be used as a quick check to see if your battery MAY be getting low. It is not a substitute for checking the battery with a DVM, it does not have the accuracy that you’re going to get with a dedicated instrument. You NEED a digital voltmeter in your tool box anyway; if you don’t have one, run out to Radio Shack and buy one, they’re under $20.

2) Balance your model with the batteries in place.

AAA and NiMH batteries are fairly heavy, as are some of the larger Lithium batteries such as a CR123. Alkaline 9V batteries are VERY heavy, almost 50 grams (nearly 2 oz!) You should always weigh your rocket and determine the CG with the intended flight battery in place. We strongly recommend using a program such as RocSim or Open Rocket to determine the flight path of your rocket with the intended payload configuration (batteries, sled, etc.) and motor.

If the rocket was designed for egg lofting or multiple deployment, chances are pretty good that just about any configuration with an Eggtimer isn't going to cause any stability problems, since those payload configurations are generally at least as heavy as a fully-loaded Eggtimer sled using a 3v battery and LiPo battery for deployment. However, if you are building a rocket from scratch or adding a payload bay onto a kit, particularly with a BT-60 or smaller body tube diameter, you may end up with an over-stable rocket that wants to weathercock into the wind or dance around on its tail after clearing the launch rod, due to excessive nose weight. Using a larger motor or one with a higher average thrust (i.e. E20 instead of E9) to get more speed off the rod will help that situation somewhat, as will using a longer launch rod. If you do the latter, you might want to increase its diameter, to keep it from "whipping" as the rocket travels along its path, or better yet use a rail launcher. If you can, you can also shorten the length of the booster to bring the CG towards the rear; you should check this carefully with a flight simulator program first before you start cutting, however!

Programming Your Eggtimer

Before you attempt to mount the Eggtimer in a rocket or fly it, you will need to know how the programming screens work. The Eggtimer can be programmed from just about any USB-compatible computer using the included USB-serial data cable. This cable has the interface circuitry imbedded in the USB connector, so it doesn't use up any board space or power on the Eggtimer. It uses a Prolific PL-2303 serial interface chip, which is recognized as a native serial port on Windows XP SP3, Windows Vista, and Windows 7, Windows 8, as well as by Mac OS X and many Linux distributions. The interface is a simple text terminal, VT100/ANSI compatible. No fancy GUI graphics, but it's fast and easy to use, and free terminal emulator software is readily available for every OS known to man. We like TeraTerm and PuTTY, but any emulator will do; however, to download flight data you need to be able to "capture" the data stream to a file, so make sure your terminal program supports this. If it doesn't, you can get around it by doing a Copy of the text that gets output to your screen, and doing a Paste into a text editor such as Notepad.

If you do not already have a suitable terminal program, or you need the USB-Serial driver for the cable, go to www.eggtimerrocketry.com, click on the Links tab, and you will be directed to a suitable download location. Note that we ***try*** to support non-Windows operating systems since the Eggtimer is OS-agnostic, however we're Windows geeks so we can't guarantee that we've had a chance to try every possible configuration (especially with Linux!)

Before you try flying your Eggtimer, we recommend that you play around with the programming screens on your workbench, until you are comfortable with it. You can't hurt anything, and if you totally mess up the memory you can always do a Master Reset to clear out all of your experimental values.

To get to the Eggtimer menu screen, be sure that the power on the Eggtimer is OFF. Connect the GND, RX, and TX connectors of the USB-serial cable to your Eggtimer, being careful to match them up to the terminals so marked, then plug the USB end into a USB port on your computer. Launch your terminal emulator program, configure the serial port for 19,200 baud, and connect to the serial port. Power up your Eggtimer, and you will see this screen. You will also hear a beep-beep sound to let you know that you're in programming mode. When you do, hit the Enter key a few times, this will let the Eggtimer know that you have a terminal connected and you're not ready to start a flight. Note: Be sure to hit Enter within 30 seconds of the beep-beeps, or it will go into flight mode and you'll have to reset it again to get to the programming screen.

```
Eggtimer Build_1.44             Flight Settings
                                                             T:23.50
    A    100   LDA <50-500 by 50>                            P:97804.00
    B    10    Burn/Coast Samp/sec:  <4,5,8,10,15,20,25,33>     A:975.39
         100   Interval (ms)                                 Main:0
    C    2     Descent Samples/sec <1,2,4,5,10>              CHB:0
         500   Interval (ms)                                 Batt:3.12
    D    0     Main Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW>
    E    300   Main Altitude <100-2000 by 50, 0 @ Apogee>
    F    1     Main ON-Time <1-9,0-Cont.>
    G    0     CHB Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW, 4-Airstart>
    H    1     CHB ON-Time <1-9,0-Cont.>
    I    1     Burn Timer <1-20>
    J    0     Airstart Delay <0-9>
    K    0     Telemetry


    O          Comments:



                   Options: Esc, A-O, ?
```

Options are selected by hitting the letter in front of the value (A-O), then using the + and – keys to toggle the values up and down. The Esc key (usually in the upper left corner of your keyboard) exits the screen and saves the flight settings, you will hear a long beep that confirms the save and then the Eggtimer will halt (both lights will blink continuously). The ? key allows you to download saved flight data (more on that later). Finally, there are a few "hidden" keys for special functions, such as erasing all the flight memory and setting the flight settings to defaults, these are covered in the Appendix.

Note that if you do not hit ANY keys or press the button down to select a pre-programmed flight profile, the Eggtimer assumes that you don't have a PC connected and you are ready to start a flight. The beep-beep that tells you that you are in programming mode last for 30 seconds, then the Eggtimer begins the flight sequence. For this reason, if you ARE programming it you should hit Enter a few times to let it know that you have a terminal connection; once you do, it will terminate the flight sequence and save your settings and halt after you are done programming.

Note: When you first get your Eggtimer, the flight memory is NOT initialized. Please see Appendix A for instructions on how to clear the flight memory by performing a Master Reset. You will need to do this before you can program or fly the Eggtimer, or unpredictable results may follow!

Field Descriptions:

A Launch Detect Altitude <50-500 by 50>

Also called LDA, this sets the altitude Above Ground Level (AGL) that the Eggtimer uses to detect that a valid flight is in progress. It can be as low as 50' or as high as 500'. It does several things:

1) Provides a start-of-flight reference point that other events are compared to.

2) Starts the flight recording sequence to "regular" memory. Prior to reaching LDA, altitude readings are saved into a special "rotating" pool of memory that gets overwritten after 50 samples.

3) Starts analyzing the velocity profile to detect burnout.
]
4) Starts the Burn Timer.

The "correct" value of this setting depends on several factors:

1) The thrust-to-weight ratio of your rocket. Higher ratios will generally call for higher settings; slower moving rockets may require a lower setting.

2) The Burn Sample Rate. Higher sample rates may require a higher LDA setting.

3) The temperature, time of day, and the wind. Yes, these are actually factors. Altimeter chips are sensitive to changes in barometric pressure, so any extraneous factors that would cause the pressure to change will have an effect on the recorded pressure/altitude reading. If it's a windy day, or if the sun is starting to come up or set then the temperature will change more quickly, and the base altitude reading may be thrown off if the rocket sits on the pad for too long. To compensate for that, the Eggtimer takes five baseline altitude readings, then averages in a new reading every 15 seconds while it's sitting on the pad. If the rocket is sitting on the pad for an extended period of time, the baseline temperature and pressure will be more accurate than if the first samples were used.

We recommend that you use the default value of 200' until you get some flights under your belt, or if you know that burnout will be at a lower altitude than this (if so, however, your rocket is most likely underpowered!) Ideally, this setting should be significantly less than the burnout altitude of your motor (one more reason to use a modeling program!) Note that some functions (such as burnout detection) require the altitude to be above the LDA setting before they will be triggered, so if you are in doubt you should set the LDA a little lower.

B Burn/Coast Samp/sec. <4,5,8,10,15,20,25,33>

This selects the number of samples per second that are written to the flight memory during the Burn and Coast portions of the flight, which begins at Launch Detect and ends at nose-over. This value is also used for the pre-launch sampling which occurs prior to Launch Detect, while the rocket is either sitting on the pad or is approaching LDA. Values are 4,5,10, 15, 20, 25, and 33 samples per second.

Note that there is a finite amount of memory available for each flight, and using higher sample rates will consume more memory and may not necessarily produce a more “accurate” flight graph. In fact, if you have a rocket with a relatively low thrust-to-weight ratio, you are much better with a lower sample rate, i.e. 4, 5, or 8 samples per second. If you have a mid-power rocket with a thrust-to-weight ratio of 10+, a sample rate of 10, 15, or 20 will give you the best graph. If you have a screamer with a 20+ thrust-to-weight ratio, then it’s OK to use 25 or 33, and you will get the best mach-transition detection with those rates for such a beast (assuming that it will go to Mach 1+…)

Be careful not set the sample rate too high. Besides wasting memory, the noise due to variations in air pressure will be proportionately higher for smaller interval sizes. If you choose too high a sample rate, you will most likely see a lot of noise in the velocity graph.

We recommend that you set this value equal to the average thrust/weight ratio of your rocket. For example, if you have a rocket that weighs 18 oz, that’s 5N (3.6 oz = 1 N). If you are flying on a G77 motor, the thrust/weight ration would be 77N/5N, or about 15/1. In this case, you should use the 15 samples/second setting to produce a good graph and enough samples to produce an accurate burnout detection.

Thrust-to-Weight Calculation:

A Weight in oz……………………….

B Divide A by 3.6 (Weight in N)....…

C Average thrust of Motor (N)..……..

Thrust/Weight Ratio (C/B)………..

Burn/Coast Interval (ms.)

This is the amount of time, in milliseconds (1/1000 sec.) represented by the Burn/Coast Sample setting. For example, if you set the Burn/Coast Sample setting to 10 samples/sec, each sample will be 100 ms. You cannot change this setting, it is automatically populated when you change the Bur/Coast Sample rate.

C Descent Samples/sec. <1,2,4,5,10>

This selects the number of samples per second that are written to flight memory after nose-over is detected, until the rocket lands and sampling ends. Values are 1,2, 4, 5, and 10 samples per second.

Nose-over is defined as one second after apogee, the maximum altitude. Since apogee can't be predicted ahead of time, it must be inferred after readings for altitude begin to drop. The Eggtimer does this after one second of descending readings after the last maximum altitude reading, on the theory that if that much time has passed then you're not going any higher. Since the rocket's vertical component velocity at apogee is by definition zero, it will be moving relatively slowly at nose-over, so if you choose to deploy a parachute at that time you won't put much strain on your recovery device or suffer the dreaded "zippered" body tube (unless your rocket happens to be going horizontal, of course…)

Descent sampling rates can be very low, because the rocket is going to be descending very slowly, typically under 50 ft/sec on a drogue parachute and 20 ft/sec on a main parachute. It doesn't make sense to waste memory by recording very small differences in altitude change. At 50 ft/sec, and a sample rate of 10 samples/sec, the rocket is only moving 5 feet per sample. If you're descending from 2,000 feet, that's 400 samples that basically look alike, so it really doesn't produce a "better" graph by turning this up, and you will either waste a lot of memory or you will run out of memory. A sampling rate of 2 samples/sec is adequate for most flights. In fact, if you're going over 5,000 feet then you should probably set it to 1 second since you probably are going to be using a higher main deployment altitude with a higher altitude flight anyway, and the difference in main deployment altitude will be minimal.

The Eggtimer will sample altitude at the descent sampling rate until either landing is detected, or it runs out of memory for that flight. Landing is detected when altitude readings of under 30 feet AGL are recorded for at least five seconds. After five seconds, the implication is that you must be on the ground, so recording stops and the Eggtimer begins beeping out the apogee. You normally will not run out of flight memory, even with a high-power rocket and higher sampling rates; however, if you do it will only stop the recording of the altitude readings to flight memory, it does NOT affect parachute deployment, so you may not even know that it happened until you dump the data afterwards. If you are unlucky enough to have your rocket get stuck in a tree higher than 30', you almost certainly WILL run out of memory; however, it will have no effect on the flight data other than showing you how high off the ground the rocket was when it got stuck.

Descent Interval (ms.)

This is the amount of time, in milliseconds (1/1000 sec.) represented by the Descent Sample setting. For example, if you set the Descent Sample setting to 2 samples/sec, each sample will be

500 ms. You cannot change this setting, it is automatically populated when you change the Descent Sample rate.

D Main Enable (0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW)

E Main Altitude <100-2000 by 100, 0 @ Apogee>

F Main On-Time <1-9, 0-Cont.> ***{only if Main Enable =1}***

F Servo PWM <0-9> ***{only if Main Enable = 2 or 3}***

This setting controls the Main (Channel A) deployment channel. When set to "0", the channel is turned off, so the other settings don't matter. When set to 1-3, the deployment channel turns on at the altitude selected by the "Main Altitude" setting, from 100 to 2000 feet or nose-over if "0" is selected. The altitude and elapsed time of Main deployment are recorded in the Flight Summary memory for downloading. Zeroes in these fields means that the Main channel wasn't enabled and therefore didn't fire.

If this setting is 2 or 3, the deployment channel sends out a pulse-width modulated stream that can control standard analog hobby servos. The amount of servo rotation is controlled by the Servo PWM value, each tick changes the servo pulse width (nominally 1.5 ms at center) by 100 microseconds in each direction, so the total possible range of servo pulse width is .5 to 2.5 ms. The servo pulses are sent for 3 seconds after deployment, this value is not configurable but is more than adequate for even the slowest servos.

The effect of the settings are:

Main Enable	Main On Time	
0	n/a	Deployment Channel OFF
1	1-9 secs	Deployment channel activated at Main Altitude for Main-ON seconds
1	0	Deployment channel activated at Main Altitude and held on continuously until Reset

Main Enable	Servo PWM	
2	0-9	Deployment channel activated at Main Altitude in servo mode, counter-clockwise, amount of rotation controlled by Servo PWM setting
3	0-9	Deployment channel activated at Main Altitude in servo mode, clockwise, amount of rotation controlled by Servo PWM setting

Note that the AUX2 channel will "shadow" CHA if you set CHA to the igniter ("1") setting, so if you want to provide redundancy in your deployment channel you can use AUX2 with a suitable buffer circuit to accomplish this. AUX2 is pre-programmed to fire for two seconds.

G CHB Enable (0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW, 4-Airstart)
H CHB On-Time <1-9, 0-Cont.> ***{only if CHB Enable =1 or 4}***
H Servo PWM <0-9> ***{only if CHB Enable = 2 or 3}***

This setting controls the Channel B deployment channel. When set to "0", the channel is turned off, so the other settings don't matter. When set to 1-3, the deployment channel turns on at nose-over, 1 second after apogee. If this setting is 2 or 3, the deployment channel sends out pulses that can control standard analog hobby servos. The amount of servo rotation is controlled by the Servo Skew value, each tick changes the servo pulse width (nominally 1.5 ms) by plus-minus 100 microseconds, so the total possible range of servo pulse width is .5 to 2.5 ms. The servo pulses are sent for 3 seconds after deployment, this value is not configurable but is more than adequate for even the slowest servos.

If CHB Enable is set to 4, the channel is used for airstarts. Channel B will fire on the way UP, after the Breakwire has been tripped AND LDA has been reached AND the Burn Timer AND Airstart Delay timers expire. See Appendix D for more information on airstarts. WE STRONGLY RECOMMEND THAT YOU RUN A SOFTWARE SIMULATION ON ANY AIRSTART CONFIGURATION BEFORE SETTING THE AIRSTART DELAY VALUE.

The effect of the settings are:

CHB Enable	CHB On Time	
0	n/a	Deployment Channel OFF
1	1-9 secs	Deployment channel activated at Nose-Over for CHB-ON seconds
1	0	Deployment channel activated at Nose-Over and held on continuously until Reset
4	1-9 secs	Deployment channel activated after expiration of Burn Timer/Airstart Delay for CHB-ON seconds, IF the breakwire has been tripped
4	0	Deployment channel activated after expiration of Burn Timer/Airstart Delay and held on continuously until Reset, IF the breakwire has been tripped

CHB Enable	Servo PWM	
2	0-9	Deployment channel activated at Nose-Over in servo mode, counter-clockwise, amount of rotation controlled by Servo PWM setting
3	0-9	Deployment channel activated at Nose-Over in servo mode, clockwise, amount of rotation controlled by Servo PWM setting

The time and altitude of the CHB deployment are recorded in Flight Summary memory for later downloading. Note that in the case of Airstarts at high speed, this data may not be accurate, since aerodynamic effects may compromise the accuracy of the pressure-derived altitude reading. Zeroes in these fields means that CHB didn't fire, either because it wasn't enabled or because the conditions to fire CHB weren't met (i.e. Airstart Delay expired after Nose-Over).

I Burn Timer <1-20>

The Burn Timer is used to indicate to the Eggtimer what the burn time of your motor is. It is used to arm an airstart ignition on Channel B, and is also used to start the velocity profile testing in order to prevent premature parachute deployments during high speed flights. The burn timer begins when the LDA is reached and expires after the selected number of seconds have elapsed from LDA. Technically, of course, the motor starts burning some time before reaching LDA while the rocket is on the pad, but for our purposes this is accurate enough since the time from ignition to LDA is typically one second or less.

In general, the Burn Timer value should be set to the rounded-up burn value of your motor. For example, if your motor burns for 2.4 seconds, you should set the burn timer to 3 seconds. If your rocket will be approaching supersonic speeds, we recommend that you never set this value less than the actual burn time of your motor.

Expiration of the Burn Timer and the altitude at the expiration of the Burn Timer are recorded in the Flight Summary memory for downloading.

J Airstart Delay <0-9>

The Airstart Delay is used ensure that the second stage igniter doesn't fire until the first stage motor has burned out. Used in conjunction with the Burn Timer, it allows you to delay the ignition of the second stage motor by 1-9 seconds, or have it fire when the first stage motor burns out (0 seconds). It has no effect if the CHB Enable setting is not 4 (Airstart mode).

There is a debate in the hobby rocketry community over whether it is "better" to have the second stage fire as soon as the first stage fires (to achieve maximum velocity) or to hold it off for a few seconds and have the rocket coast awhile first (to achieve more altitude since there will be less aerodynamic drag when the motor fires). That's the fun part of rocketry… you're free to try things for yourself and see how it works!

Some igniters take a significant amount of time to start composite-propellant motors, so this time must be considered. If your igniter takes about 2 seconds to start the second-stage motor, you will need to take 2 seconds off the Airstart Delay value, if you make it zero then the airstart will be delayed by the time that the igniter takes to fire your motor. Black powder motors (like the Estes motors) igniter almost instantaneously, so you don't need to worry about it with them. Note that if the Burn Timer expires during the mach transition phase, the Channel B igniter WILL fire if Channel B is in airstart mode. Unlike the Main deployment channel, or Channel B in drogue mode, airstart ignition is not "held off" during mach transition. This is intentional, because if you have a very fast rocket and you set the Burn Timer and Airstart Delay so that it fires during supersonic flight the assumption is that you wanted to do this.

Also note that if you set the Airstart Delay too long (way too long...) and it expires AFTER Nose-Over is detected, the CHB channel will not fire. This is a safety feature, so you don't shoot off the second stage while it's pointing downward. If you are using the second stage's motor for drogue or main chute deployment, obviously those deployments aren't going to happen; if you are using the Eggtimer's Main channel for deployment, it WILL fire, hopefully saving your rocket so you can try it again later with more accurate settings.

Airstart firing altitude and time is recorded in Flight Summary memory as CHB, since CHB is used for airstarts. Zeroes in these fields means that CHB didn't fire, probably because either the breakwire didn't open up or your delay was too long and Nose-Over occurred first.

K Telemetry <0=Off, 1=On>

When set to "1", flight time and altitude samples are sent through the serial port in real-time during flights. Data streaming begins as soon as the pre-launch sampling begins, and continues until the flight ends. Even if the memory runs out, the telemetry data will continue to stream after landing is detected, it just won't be saved to flight memory. If you're using some kind of RF telemetry package, this insures that you're going to get a continuous data stream that you can use to help find your rocket on the ground. We've all wandered around deserts, cow pastures, or farm furrows for a few hours looking for a wayward rocket, so you can see the value in this!

Only two wires are required to send data to an external device, the GND and the TX lines. TX is a TTL-compatible output (low < .8v, high > 2.4v) NOT an RS-232 output (low > 3v, high < -3v). Data is streamed at 19,200 baud, so you will need to make sure that your radio device is configured for that rate.

Note that many radio telemetry devices require a FCC Amateur Radio Novice license, so check before you purchase anything. If you want to get into telemetry, you should probably go ahead and get a Novice license anyway, you'll need the knowledge to get your telemetry system working and it's pretty cheap… the exam fee is only $14 and the study guide is about $30. Obviously, telemetry is a detailed subject in itself, so we will not cover it any further here.

O Comments

The Comments field is a 42-character free-form text field that you can use to identify you flight, enter the data, rocket name, motor, etc. We recommend that you change it for every flight to make it easier to identify the flights when you go to download them later. If you choose a pre-programmed flight profile, it will be set to "Default Flight <n>", where <n> is the flight profile number from 1 to 8.

Note that if you abort a flight, for example you reset the processor after the Ready signal begins to sound, all of the settings including the Comments will be carried over to the next flight. If you want to change the Comments, you will need to go into the programming screen before the next flight.

Pre-Programmed Flight Profiles

The Eggtimer has 8 pre-programmed flight profiles, in addition to the programming screen. These are designed so that if you are out in the field and you want to set it up with a fairly standard flight profile you can do it without having to use a computer. They are accessed by holding the button down while the Eggtimer is beeping during the programming phase. As you continue to hold down the button, you will first hear a long tone that confirms that you have entered the pre-programmed flight selection mode, then you will hear short beeps that will count out the flight program numbers in succession. When it beeps out the program that you want to use, simply release the button. It will give you a long beep to indicate that it has confirmed your selection, then it will beep out be program number before it halts. As with any menu function, you will need to Reset the Eggtimer afterwards in order to begin a flight.

The pre-programmed flight profiles look like this:

Profile No.	LDA	Burn/Coast Samples/Sec	Main Channel	CHB Channel
1	200	10	Off	Off
2	200	20	Off	Off
3	200	10	Nose-Over	Off
4	200	20	Nose-Over	Off
5	200	10	500’	Nose-Over
6	200	20	500’	Nose-Over
7	200	10	800’	Nose-Over
8	200	20	800’	Nose-Over

1 – Flight Recording Only, no deployment, low-medium power
2 – Flight Recording Only, no deployment, medium-high power
3 – Single Deployment at Nose-Over, low-medium power
4 – Single Deployment at Nose-Over, medium-high power
5 – Multiple Deployment, Main at 500’, low-medium power
6 – Multiple Deployment, Main at 500’, medium-high power
7 – Multiple Deployment, Main at 800’, low-medium power
8 – Multiple Deployment, Main at 800’, medium-high power

Note that the Comments field will contain the message “Default Flight <n>”, where <n> is the number of the flight profile.

You cannot and/or should not use the Pre-Programmed Flight Profiles for any of the following:
LDAs other than 200 feet
Burn/Coast sample rates other than 10 or 20 samples/sec
Descent sample rate other than 2 samples/sec
Servo deployment on either channel
Airstarts on Channel B, since you can’t change the Burn Timer or the Airstart Delay
Entering your own text into the Comments field

Flying Your Eggtimer

Once you've programmed your Eggtimer, you are ready for a flight. In general, a generic flight prep procedure for a multiple deployment rocket looks something like this:

At Your Prep Table
Check ALL flight batteries with a DVM and replace/recharge as necessary
Double-check your Flight Settings screen to make sure that all values are correct
Hook up any deployment charges with the computer and deployment batteries **disconnected**
Connect deployment charge shunts, if used (optional but recommended)
Confirm that the power switch, if used, is OFF
Confirm that any deployment power switches are OFF
Confirm that the Reset switch is ON

At the RSO Table
Prove to the RSO that the deployment channel(s) are disabled, and that no batteries have been connected yet (along with showing them that your rocket is flightworthy, of course!)

At the Pad
DOUBLE-CHECK that the Reset switch is ON
Connect the flight computer battery
Connect the deployment batteries
Button up the payload bay
Install and connect the motor igniter (it would be embarrassing if you forgot, huh?)

Just Prior to Launch
Turn ON the DEPLOYMENT power
If you are using the ALED/BLED LEDs and they don't come on, turn the deployment power OFF and figure out what's wrong before proceeding
Turn ON the flight computer power
REMOVE any shunts that you may be using
Turn OFF the RESET switch and step back several feet for safety
When you hear the "Ready to Launch" chirping tone, you're good to go
Move to the safety area, and watch your creation blast into the sky as the LCO hits the button

If You Get an Error Tone:
Carefully approach the rocket
Count the number of beeps to see what caused the error (there may be more than one error)
Turn ON the RESET switch to put everything on hold
REPLACE any shunts that you have installed
TURN OFF the deployment battery power
TURN OFF the flight computer power
Disconnect the motor igniter from the launcher

Take the rocket off the rod/rail, open up the payload bay, disconnect all batteries, and fix the failed condition. Then, you can start all over having hopefully learned one more thing to check in the future…

You can see that if you're doing multiple deployments, it's a lot more involved than just stuffing some wadding and the parachute into the tube, popping in the motor, and hooking up the igniter. Multiple deployments require discipline in order to make them work reliably; we've seen way more than our share of failed deployments, on everything from a small mid-power E-size rocket all the way up to an M-sized 200 pound beast. We've seen deployments fail with the top-end flight computers and with the low-cost units, even with redundancy. The reality is that most deployment failures are not the fault of the electronics; it's usually something mechanical like a bad connection, a nose cone that's too tight, etc. Having your blood, sweat, and tears free-fall from a few thousand feet and bury itself in six feet of dirt is going to ruin your whole day.

We STRONGLY recommend that you use a checklist every flight so that you don't forget anything. We have some on our website under the Downloads section (at www.eggtimerrocketry.com/downloads) or you can create your own… the main thing is that you NEED to use one. There are also plenty of other resources online, Google is great for finding them. We also recommend that you get a copy of the book Modern High Power Rocketry , it's full of good information too.

The Eggtimer Flight Sequence

At the Pad…

As you know by now, when you first turn on the Eggtimer you get a long "I'm running" beep followed by 30 seconds of short beep-beeps to indicate that you're in programming mode. When you're at the pad and the programming beeps stop, that's when the fun begins…

At that point, the Eggtimer looks at your flight profile to determine what deployment channels to test, whether it needs to test the breakwire, and it does some internal tests as well. If everything is OK, it goes on to the "Ready for Flight" sequence, indicated by a 10 second long beep. When the Ready for Flight beep stops, you will hear continuous chirping that indicates that altitude samples are being taken and saved to the pre-LDA memory, so the rocket may be launched at any time. If the Telemetry feature was enabled, elapsed time and altitude data will start streaming to the data port, and if you have an RF link you will see that data on your monitor.

If the Eggtimer fails a self-test, you will hear a high-low beep tone that counts out the error code instead of the "Ready for Flight" tone. This means that the Eggtimer has aborted the flight due to a hardware problem, AND YOU SHOULD NOT LAUNCH. Deactivate the Eggtimer, disconnect the batteries, disconnect the motor's igniter, and correct the error before attempting the flight again.

The beep codes are:

Number of Beeps	Problem
1	Battery Low (< 2.8v)
2	Altimeter Sensor or Hardware Error
3	Breakwire open (airstarts only)
4	Channel A (Main) error - igniter/servo/battery
5	Channel B (Drogue/Airstart) error – igniter/servo/battery

Note that if you are using a servo for deployment, you MUST connect a 39 ohm 1/4W resistor across the deployment channel's output terminals in addition to connecting the servo's control input to the "+" deployment channel output and the servo's power leads to the deployment battery. If you do not hook up the resistor, there will be no continuity detected across the output channel leads, AND the servo's control input has no way of being pulled to ground so it will not work.

OK, we're ready, so press the button already! ☺

In the Air…

Once your rocket is launched, altitude samples are taken at a rate determined by the Burn/Coast Samples setting. These setting are saved into a special rotating pool of memory until the LDA altitude is reached, then they begin to be saved into the "regular" memory. When your rocket reaches LDA, the burn timer starts and velocity profile data begins to be computed.

Velocity readings are computed based on the difference in altitude between successive samples and the elapsed time between the samples. In a perfect world, this would give you a precise and 100% accurate velocity reading. The reality is somewhat different, however. There are several things that can introduce errors into the velocity reading: angle of attack, wind, errors in the pressure readings due to aerodynamic influences, and time differences. If you measure the launch angle prior to launch, you can calculate this error, although anything (such as wind) what makes the rocket deviate from a straight-line path will pretty much negate this compensation unless you had set the angle to a really high value (over 15°). If the pressure ports in your payload bay aren't sized properly, this can introduce an error, particularly if they're too big and you have two of them opposite each other (you'll get a crossflow through the payload bay which makes the pressure readings very noisy). Finally, differences in the processor's timing may introduce errors, although the readings are taking at relatively precise intervals so it's going to be very small.

The good news is that the magnitude of these errors tend to be proportionate to velocity as the rocket ascends, so they respond well to being filtered with mathematical noise filters. We use a variation of a Kalman filter to smooth out this "noise", so burnout and transition to supersonic speeds and back can be detected with relatively good accuracy.

As the samples are taken, maximum values for altitude and velocity are logged, at the rate determined by the Burn/Coast samples/sec setting.

Mach Transition…

As your motor continues to burn and the velocity increases, if the velocity exceeds 800 ft/sec aerodynamic shock wave buildup can fool the pressure sensor into thinking that the rocket is descending when in fact it is actually ascending at a rather rapid speed. If this were not taken into account, the flight computer might deploy parachutes at near-mach speed, which would undoubtedly break something and ruin your day, not to mention what an object falling from the sky at these speeds could do.

To prevent this from happening, the Eggtimer uses a predictive mechanism to hold off deployments until it's safely out of the mach "danger zone". Real-time altitude readings are run through a Kalman filter, which "smooths" the noise from the pressure readings. The smoothed readings produce a much gentler velocity profile, which allows it to be used to obtain reasonably accurate velocity samples. If the velocity reaches 500 ft/sec, it is assumed that the rocket may reach Mach speeds, and deployments are inhibited so that the sudden pressure spike (and perceived loss of altitude) does not result in a premature deployment. When the velocity drops below 100 ft/sec (presumably near apogee) for at least 1.5 seconds, deployments are enabled.

Note that if Channel B is configured in Airstart mode, it WILL fire the igniter during mach transition. Once your rocket hits LDA, airstarts are under the control of the timers and are not dependent on altitude readings. It is assumed that you have modeled the flight and appropriately set the timers, so if you do NOT want the second stage to fire during mach transition you must set the Airstart Delay to a higher value so that the rocket will coast for a longer period of time.

Apogee and Nose-Over

Assuming that your rocket is moving more or less straight up, it will continue to slow down during the coast phase until it gets as high as it's going to go. If the rocket was going absolutely straight up, the velocity at this point would be zero; it would simply start falling to the ground. In reality, this almost never happens, because you usually angle the rod/rail at a slight angle so that the rocket takes off away from the flight line. This results in the velocity disparity that we've previously mentioned. The rocket usually has some forward velocity at apogee; hopefully it's relatively small so your parachute deployment happens at a low velocity and won't cause any mechanical problems like a broken shock cord or a zippered tube. It also means that you really don't know that you've reached apogee until ***after*** you've been there because you can't define apogee as when the altitude change reaches zero, because velocity won't be the same as altitude change if you have a horizontal velocity component.

Accordingly, the Eggtimer fires the drogue parachute at Nose-Over, which we define as one second past apogee (highest recorded altitude). If the rocket is still going up, chances are that its altitude is going to keep increasing before a one second interval elapses, so you're unlikely to get a false apogee detection. If the rocket hasn't gained any more altitude after one second then you must be starting on your way down, so the Eggtimer fires the drogue at that point, for a time determined by the CHB ON setting (or 3 seconds for a servo). Assuming that the mechanical portion of the drogue mechanism is working properly, your drogue chute will pop out and your rocket will start slowing down.

When Nose-Over is detected, the Eggtimer begins using the Descent Samples value for altitude sampling. This setting is typically very low, 2 samples/sec is good and in most cases 1 sample/sec is fine too. As the rocket descends, when the altitude drops below the Main deployment altitude the Main channel is triggered, for a time determined by the CHB ON setting (or 3 seconds for a servo). At this point your Main chute should pop out, to thundering applause and a hearty round of high-fives all around.

On the Ground

The Eggtimer detects that your rocket is on the ground when the AGL altitude is below 30 feet for over 5 seconds. Unless you launched from a hill, it's unlikely that your landing altitude will be lower than that by a full 5 seconds times the descent rate, which is typically about 20 feet/sec for a main chute. That works out to a landing altitude 100' below your launch altitude, which would be unusual.

What is more likely is that it won't be able to figure out that you've landed, generally because your rocket landed on high ground or a tree. If that happens, it will continue to record altitude samples at the Descent Samples setting, until it runs out of flight memory. When that happens, it will call it a flight.

If you run out of flight memory during an actual flight, drogue and/or main chute deployment WILL occur. These events are not dependent on recording the flight samples, they are only dependent on time/altitude and therefore are completely independent of the recording process. The recorded deployment altitude and time ARE dependent on the recording process, however, so you may not get an accurate deployment altitude/time in your flight summary download if this happens.

Once on the ground, the Eggtimer will start beeping out the maximum altitude continuously for anyone within earshot to hear. If you have the Telemetry option turned on, instead of beeping out the altitude it will continue to send out the current altitude, which should be pretty low (unless you managed to get stuck in a tree.) It also acts as a verification that you had a successful flight, but getting your rocket back in one piece pretty much told you that, didn't it?

After the Flight

Once you've noted the maximum altitude so you can brag about it to your friends, you should turn off the Eggtimer as follows:

Turn ON the Reset switch; this hold the Eggtimer
Turn OFF the deployment batteries
Turn OFF the Eggtimer

Gather your rocket up and take it back to your work table, get a can/mug/bottle of your favorite beverage, then you are ready to download the data.

Downloading Flight Data

The Eggtimer records your last 32 flights into its flight memory, reporting three types of data:

- Flight Settings – All of the settings on the Flight Settings menu (or the Presets, if you used one)
- Summary Data – Stuff like apogee, length of flight, maximum velocity, etc.
- Detail Data – A CSV-formatted data stream with elapsed time, altitude, and velocity

You may not be able to use all 32 flights' worth of data, because if you abort a flight you are going to get a "null" flight that consists only of the Flight Settings with no data. You will also get this if you have it on the bench and pass through the programming screens. This is not necessarily bad, you can use it while your rocket is on the pad to see if the deployment channels pass the self-test, then you can turn on the Reset switch until about a minute before launch. Some clubs launch only at the hour or half-hour, so you can save some battery life by checking the deployment channels first then holding on the pad until right before your launch slot.

To enter the Flight Display screen, hit the "?" key at the Flight Settings screen. You will see something that looks like this:

```
Eggtimer Build-1.44     Saved Flight Summary

    Page      Apogee    Max V    Comments
A   18        0         0        New Flight
B   17        1235      420      F30 Flight 3
C   16        0         0        F30 Flight 3
D   15        1346      416      F30 Flight 2
E   14        0         0        F30 Flight 2
F   13        0         0        Test F30
G   12        1170      387      Default Flight6
H   11        0         0        Test
I   10        642       239      Vagabond E12
J   9         939       412      Vagabond E28
K   8         1509      487      Vagabond F39 #2
L   7         1498      512      Vagabond F39 #1
M   6         509       193      E9
N   5         627       267      E12
O   4         2131      442      Goblin G77 #2
P   3         2281      458      Goblin G77 #1
```

The most recent 16 flights are displayed on the first screen. To toggle between the two 16-flight display screens, hit the "+" or "-" key.

To select a flight to display, hit the letter key for that flight, from A-P. (Case doesn't matter, you can hit "a-p" too). For example, if you wanted to dump the data for the flight "flight 2012-09-08c" you would hit the "D" key. From this display, you can see that there are some null flights; since some of them have the same name as the subsequent "real" flight, you can probably surmise that they were deployment channel tests before a live flight.

After you hit the letter for the flight, you will be prompted to hit the button to start the download. At this point, you should go into your terminal program and select the appropriate option for dumping the text to a file. Since the summary data and the detail data are dumped in two separate operations, you can save them separately by simply closing the buffer after dumping the summary data, then opening it again for the detail data.

We recommend that you use a file name like “Flight 2012-08-26 No1.txt” for the summary data, and “Flight 2012-08-26 No1.csv” for the detail data. This way, when you open up the data files you will automatically associate the data type with the right program, i.e. Notepad (or Notepad++ if you have it… if you do not, we recommend that you get it; you’ll never use Notepad again!) for the summary data, and Excel for the CSV-formatted detail file. Be sure to use “.csv” as the extension for the detail data if you want Windows to recognize it as a CSV file when you open it.

For example, here is some actual flight data, taken from an Estes Vagabond that was modified for multiple deployment. (We really like this rocket, it’s an easy hack and it’s cheap so if you wreck it you’re not out much). The rocket’s weight was just over 3N ready to fly, we used an Aerotech 24mm 24/40 RMS F24-7 reload with the motor’s ejection charge being used for the drogue (a 12” plastic chute with a big spill hole) and a “pistol powder” ejection charge set to fire the Main at 300’ (an 18” nylon chute). We used 8 samples/second for both the Burn and Coast samples (derived by taking the motor’s 24N average thrust and dividing by the weight of the rocket, 3N, giving 8 samples/sec), and 2 samples/second for the Descent. The Burn Timer was set to 2 seconds.

When you press the button to download the summary data, you will see something like this. Note that all times are in milliseconds, referenced from the t=0 sample (the last sample in pre-launch memory with an altitude <= 0).

```
Flight Summary:

Eggtimer  Build-1.44    Flight Settings
...Comments:...Flight 2012-08-11b
...Launch Detect Altitude <50-500 by 50>...200
...Burn/Coast Samples/sec:  <4,5,8,10,15,20,25,33>...8
...Interval (ms)...125
...Descent Samples/sec <1,2,4,5,10>...2
...Interval (ms)...500
...Main Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW>...1
...Main Altitude <100-2000 by 50, 0 @ Apogee>...300
...Main ON-Time <1-9,0-Cont.>...2
...CHB Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW, 4-Airstart>...0
...CHB ON-Time <1-9, 0-Cont.>...2
...Burn Timer <1-20>...1
...Telemetry...0

ASL Altitude: 557
Temperature: 3840
Beep Code: 0
Apogee:1215
Time@Apogee:7375
Max Velocity:280
Time@ Max Velocity:1875
Flight Timer:49000

Altitude Burnout:248
Time@ Burnout:2000

Altitude Burn Timer:391
Time@ Burn Timer:2625
```

```
Altitude Nose-Over:1214
Time@ Nose-Over:8375

Altitude Main Trigger:298
Time@ Main Trigger: 75100
```

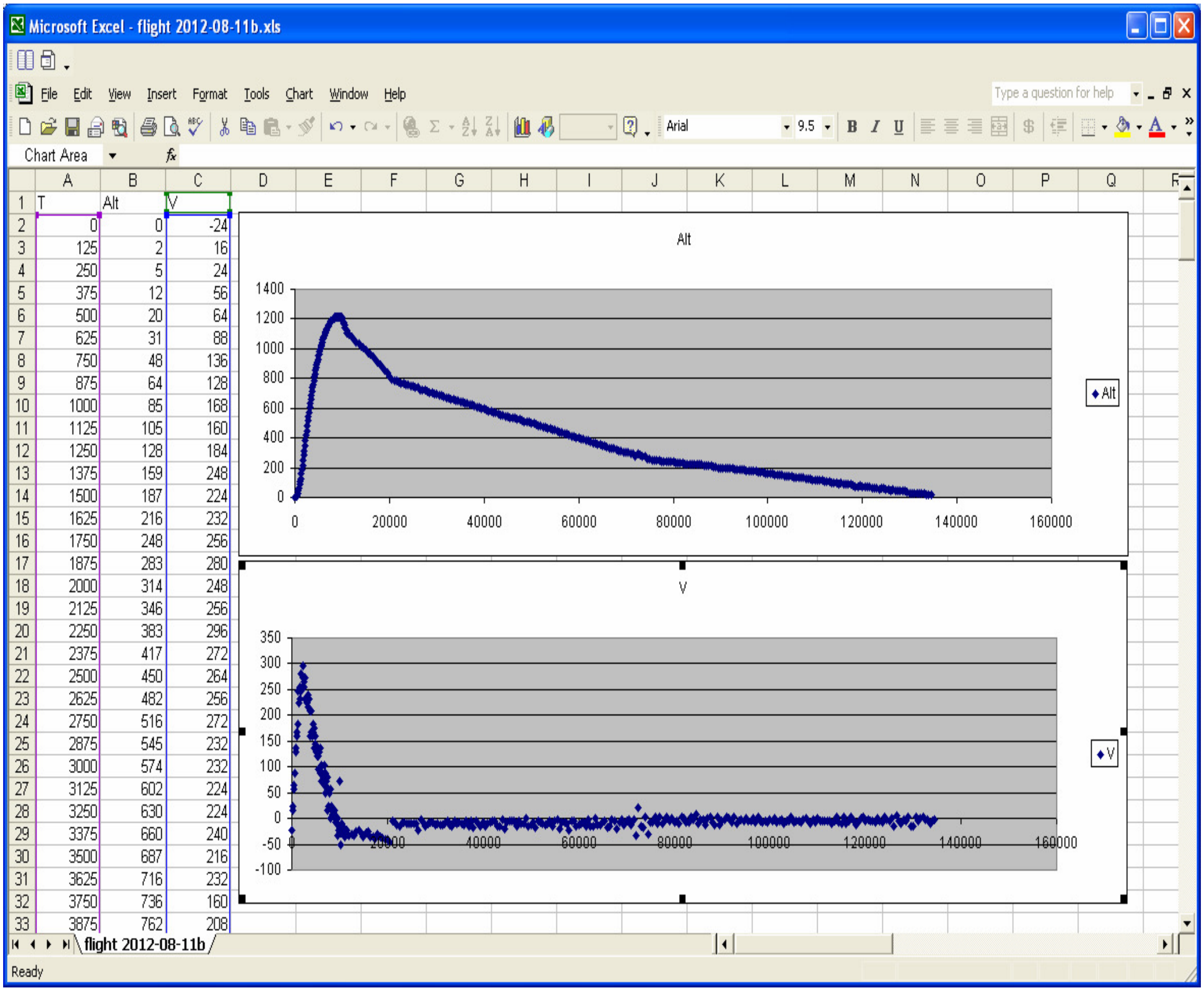
As you can see from this data, the rocket got to 1215 feet, had a maximum velocity of 289 feet/sec.

To dump the detail data, press the button again… you will see something like this.

```
T,Alt,V
0,0,-24.00
125,2,16.00
250,5,24.00
375,12,56.00
500,20,64.00
625,31,88.00
750,48,136.00
875,64,128.00
1000,85,168.00
1125,105,160.00
1250,128,184.00
1375,159,248.00
1500,187,224.00
1625,216,232.00
1750,248,256.00
1875,283,280.00
2000,314,248.00
etc..
```

The first data field is the elapsed time in milliseconds, the second data field is the altitude in feet, and the third data field is the velocity in ft/sec. After you close the file, you can open it with the data analysis program of your choice. For example, if you import it into Excel, you can graph Time vs. Alt and Time vs. Velocity. See our web site, www.eggtimerrocketry.com , for step by step instructions on how to do this; it's not difficult once you get the hang of it. You can also import the CSV data into statistical software packages, although this is beyond the scope of what we're doing here.

An Excel graph from this data looks something like this:



	A	B	C
1	T	Alt	V
2	0	0	-24
3	125	2	16
4	250	5	24
5	375	12	56
6	500	20	64
7	625	31	88
8	750	48	136
9	875	64	128
10	1000	85	168
11	1125	105	160
12	1250	128	184
13	1375	159	248
14	1500	187	224
15	1625	216	232
16	1750	248	256
17	1875	283	280
18	2000	314	248
19	2125	346	256
20	2250	383	296
21	2375	417	272
22	2500	450	264
23	2625	482	256
24	2750	516	272
25	2875	545	232
26	3000	574	232
27	3125	602	224
28	3250	630	224
29	3375	660	240
30	3500	687	216
31	3625	716	232
32	3750	736	160
33	3875	762	208

You can clearly see that the rocket hits peak altitude at about 10 seconds, and maximum velocity at about 3 seconds. Note that the velocity graph is jaggedy compared to the altitude graph; this is due to the aforementioned issues with deriving velocity from barometric pressure. The important thing is that you can see clearly where the velocity peaks then drops sharply; this is where the motor burned out and aero drag immediately started to slow the rocket. You can also see where the motor's ejection charge fired about a second after apogee (the 7 second delay was a good fit), and the rocket starts to slow. In this case, the drogue chute took a few seconds to open up, so you see that the rocket descended about 300 feet in 8 seconds, then slowed as the drogue chute opened up, then slowed even more at around 75 seconds at a little under 300 feet as the main chute opened. The last 250 feet or so took nearly a minute; obviously both the drogue and the main chute were too big, but this was a very conservative build. If you look at the velocity graph, you will see disturbances where the ejection charges fire; this is because the jerk of the parachute coming out causes a short pressure "bounce".

Building Your Rocket to Use the Eggtimer

Mounting

In order to use your Eggtimer with a rocket, it will need to be properly mounted and secured. The Eggtimer is designed to be used with just about any rocket with a 1.35" dia. (BT-55) or larger payload bay, so any rocket that will take an egg will easily take the Eggtimer. However, there is some preparation that will be need to be done to secure it properly. If you're familiar with egg-lofting, you know that you can't just throw an egg in the payload bay and expect it to come back unharmed. The Eggtimer is no exception, like an egg it needs to be properly protected from the forces of flight.

The easiest way to do this is to build a "sled" for it, a piece of 1/8" plywood or basswood that can simply slide into the payload bay. If properly done, a sled can be easily removable for post-flight downloading. Better yet is to build a payload bay out of couplers that fits into the rocket between the booster and the main parachute sections, there are some excellent references for constructing payload sections in the Appendix.

The minimum size sled that will work with an Eggtimer is 4" long. We recommend, however, that you make the sled as long as the payload bay will allow; if you have a 6" payload bay, for example, your sled should be the entire 6" in length. You do not want the sled moving around in the payload bay during flight!

Mount the Eggtimer with at least two, preferably four, 4-40 screws and 1/4" nylon spacers, and use nylon washers on the top of the board if you are using metal screws. You can use self-tapping screws if the sled is wood, but we recommend that you put a dab of CA glue in the holes to keep the screws from coming loose. If you use machine screws, put a dab of Loctite® on the threads before you install the nuts, or use Nylock-type nuts. You can also use nylon screws, they are a little lighter but they're more expensive too. NEVER use Velcro to mount your Eggtimer, it will generate static electricity when you remove it and you may damage your Eggtimer permanently.

Some Mounting Hints

Batteries can be mounted in several ways. Small zip ties work well, so does double-sticky foam tape for LIGHT batteries (like 80 maH LiPo batteries, for example). If it's a rechargeable battery and you don't expect to remove it, a dab of epoxy works too. Do NOT use double-sticky Velcro, it WILL come loose in flight and is a magnet for static electricity.

No matter how you mount the batteries, make sure that they can't work loose during flight. In the case of coin cells, we recommend a holder with a positive closure; the "slide-in" holders are asking for trouble. If you DO have to use a slide-in holder, make sure that the open side faces horizontally, and use a zip tie or a few wraps of tape to ensure that the battery can't slide out.

Two AAA batteries (or AAAA if you can find them) work very well, however the holders can be a problem because they are usually spring loaded and can come loose in flight. We do not recommend using AAA batteries unless the body tube is large enough to mount them horizontally, so that the G forces of launch and deceleration cannot make the batteries shift and lose contact. Even then, we recommend that after you install the batteries in the holder you wedge a thin piece of wood (scrap from the sled is good) in between the spring loops to take up any slack. You should ONLY use completely enclosed holders, such as Radio Shack #270-414, and you should use at least TWO zip ties to mount it: one vertically and one horizontally. Alternatively, you can also use a dab of epoxy to mount the battery holder, but you still need to have at least one zip tie to keep the battery door from popping open during flight.

Multiple Deployment Hints

1) ALWAYS weigh your completed rocket with all batteries and the motor that you plan on using, determine the CG, and run it through a program like RocSim or Open Rocket to check the stability and predicted flight path of your rocket. Google to see if there is a RocSim file for your rocket, and/or modify it accordingly if you have made changes or have a custom build.

Yes, this IS rocket science…

2) Similarly, use RocSim/Open Rocket to determine the best delay time for the drogue deployment. Ideally, it should be within +/- one second of predicted apogee, when your rocket is moving at a relatively slow speed (< 20 ft/sec is ideal). If you are going to use the motor's ejection charge to fire the drogue, and the motor has a fixed delay time (i.e. one that you cannot "drill out" to adjust), it might be a good idea to do this before you buy your motors, so you can purchase them with the correct delay time.

3) Prior to flight, perform a Ground Test procedure with a test load and your ignition battery. We recommend using miniature Christmas Tree lights, they work very well for testing with either 9v or 12v A23 batteries. You should perform at least one Ground Test with the pyrotechnic charge that you are going to be using to be absolutely sure that it will fire with the battery that you choose, and that it's strong enough to kick out your parachute. This obviously should be done OUTDOORS in a well-ventilated area. There may be local regulations that prohibit pyro testing where you may be living… check with your local fire marshal if you aren't sure. If you can't test in your backyard, check with your local club to see where you can test; they may have an area set aside during launches for such testing.

4) Batteries can be heavy, especially 9v "transistor" batteries. For best flight performance, use the lightest batteries that you can get by with. 3.7v single-cell Li-Po batteries work great for both the flight computer power and for deployment with low-current igniters like a Quest Q2G2, but you need to charge them properly and solder 24 gauge pigtails to the battery leads, since most of them are very fine and can easily break under flight stress. After you solder the pigtails, fold the pigtails over and wrap the battery with a couple of wraps of electrical tape to strain-relieve the delicate battery leads.

5) 12v “A23” batteries work well for firing low-current igniters (i.e. Quest Q2G2’s), but they don’t have a lot of capacity so they’re not going to be able to fire very many times before they will need to be replaced. 9v batteries are heavy (almost 50 grams) but they will fire a Q2G2 igniter several dozen times before you need to worry about replacing them. That being said, batteries are a lot cheaper than motors and rockets, so if there is any doubt whatsoever about the capacity of your ignition battery, replace it. When in doubt, TEST THEM FIRST before you try to fly them, and replace them if they read under 80% of their rated voltage.

A lot of people replace their deployment battery after EVERY flight regardless of the voltage or the current draw being used. With a 9v Alkaline battery and a Quest Q2G2 igniter this may be overkill, but for smaller batteries such as an A23 12v battery it may be a really good idea. Compared to the cost of the rocket, motor, Eggtimer, and last but not least, your time, a few dollars per launch to ensure that you get a good deployment is cheap.

6) The default ON-Time parameter is 2 seconds; that is long enough for just about any type of electric match or igniter that you’re likely to use. Most igniters will go electrically “open” after they fire, so they’re not going to draw any more current from your battery after they fire even if you set this value higher. If you have any doubt at all about how long it takes your igniter to heat up, increase the On-Time value; chances are that it won’t drain the ignition battery more than it needs to. If you are using a relatively high-current igniter like #36 gauge nichrome wire and a 9v battery, you may want to increase this time by a second or so to give the nichrome extra time to heat up.. If the igniter fires and does not melt through, however, it’s going to keep drawing current from the battery while it’s on, so don’t make this parameter any longer than you really need it to be or you may unnecessarily drain your battery. Of course, if you’re replacing the battery after every flight it probably doesn’t matter.

7) If you are using a non-pyro deployment (i.e. a small motor or a solenoid), you may need to set the ON-Time to “continuous” (zero). That setting will keep it on until you reset the Eggtimer, or press the button at the end of the flight to beep out the maximum altitude. If you have a high-current load, like a solenoid, it WILL drain your deployment battery. Keep that in mind before you set this to zero!

8) If your rocket is going to be approaching or exceeding mach transition, we strongly recommend that you make sure that your burn samples are set appropriately. In general, it’s going to be over 20 for a mach-capable rocket because it’s virtually by definition a high-power rocket. Be sure to set the Burn Timer appropriately.

9) If you are using a pyro ejection charge, you MUST have an external power and/or reset switch to disarm the Eggtimer until shortly before launch. This is a Tripoli and NAR safety requirement, and is just good common sense too. Additionally, you may want to put an external shunt across the igniter terminals, pulling it out right before you remove the Reset shunt. This will prevent the pyro from firing when you are setting it up on the pad. We recommend that you use a large gauge wire, at least 16 awg, for the shunt, because some of the very low current igniters such as a Quest Q2G2 will fire at under 200 milliamps of current.

10) If you are using servos, you MUST connect the “control” lead of the servo to the “+” output of the deployment channel, and you MUST connect a 39 ohm 1/4W resistor between the “+” and

"-" outputs of the deployment channel. If you leave out the resistor, the servo will not work, because the resistor pulls the control line down to ground while the channel is turned off. The Black lead of the servo will go the the "-" lead of the deployment battery, and the Red lead of the servo will go to the "+" lead of the deployment battery. Note that most servos will run fine on a 3.7v Li-Po cell, a 3.6v NiMh pack, and any battery up to 6V. You can also use a small PX28 6v battery, the Eggtimer limits the travel time to 3 seconds, so it will probably last awhile. Nevertheless, like any deployment battery, you need to check it with a DVM before each flight.

Air Pressure Ports

For the altimeter to work properly, there must be an undisturbed exposure to the outside air pressure. This is a little trickier than it sounds, because the air is whizzing by the rocket at several hundred feet per second during flight, so as you may remember from Bernoulli's principle in Physics this is going to cause the pressure at that location to drop if the pressure cannot be equalized. The trick is to make the air pressure ports large enough so that sufficient air can enter the payload bay, but not so small that it drops with air velocity. If you make the ports too large, then crossflow air currents can disturb the airflow and your readings will be very "noisy" during the boost and coast phases when the rocket is moving at high speed.

There has been a lot written about sizing air pressure ports for model rocket altimeters, but most of the literature agrees that the total area need to be about 1/500 of the volume of the payload bay. For example, if you have a 3" rocket with a 9" payload bay, that's about 63.5 square inches, so the total area of the ports should be about .12 square inches (about 1/8 square inch). We recommend that you use three ports, equally spaced around the diameter of the tube, so each port would be about 1/24 square inch, or about ¼" in diameter. You need these ports to be at least four body tube diameters from the tip of the nose cone to prevent possible pressure build-ups during high speed flight, particularly during mach transition, but this is usually not an issue with most multiple deployment rocket configurations because the payload bay is typically located behind the main parachute bay.

A quick formula for the diameter of the holes, assuming that you are using three is:

1) Length of Payload Bay in inches: ___________________

2) Diameter of body tube in inches: ___________________

3) Square of #2 ___________________

4) Multiply #1 and #3 ___________________

5) Divide by 1500 ___________________

6) Take the square root: ___________________

This is the diameter of the holes you need

For example, a small 5” payload bay 1.5” inches in diameter (inside of a BT-60 coupler…)

1) Length of Payload Bay in inches:	5
2) Diameter of body tube:	1.5
3) Square of #2	2.25
4) Multiply #1 and #3	11.25
5) Divide by 1500	.0075
6) Take the square root:	.0866

We’ll multiply this by 32 to get the size of the drill in 1/32nds, it’s 2.77/32” so we should be fine with a 3/32 drill bit. Make sure that you use a good sharp drill bit, you do not want any burrs on the outside of the tube disturbing the airflow. You may want to sand the area around the holes with some #400 extra-fine sandpaper to smooth out any burrs or flash, and possibly harden it slightly with some thin CA glue too.

Appendix A – Special Functions

There are some special functions that are accessed from the Flight Settings screen that are not on the menu. In general, you will probably not need to use these functions very often, if at all, but you need to be aware of them in case you do.

Master Reset

A master reset completely clears all flight memory, and resets the flight settings to the defaults. The default settings are:

```
Launch Detect Altitude: 200
Burn/Coast Samples/sec: 10
   Interval (ms):   1000
Descent Samples/sec: 2
   Interval (ms):   500
Main Enable:         0  (off)
Main Altitude:       300
Main ON-Time:        2
CHB Enable:          0  (off)
CHB ON-Time:         2
Burn Timer:          2
Airstart Delay:      0
Mach Timer:          2
Telemetry:           0  (off)
```

To perform a Master Reset, hold the Ctrl and Y keys down together at the Flight Settings screen, you will then see this screen:

```
 Eggtimer Build-1.44         Master Reset


                       **** WARNING ****

             DO YOU REALLY WANT TO CLEAR ALL FLIGHT MEMORY???


                      Press BUTTON for 10 seconds to Reset
```

To perform the reset, you need to hold the Button on the Eggtimer down for at least 10 seconds. When you do, you will hear a long confirmation tone, then you will see a progress screen as the memory pages are reset. Leave the Eggtimer alone until the process is complete. When the reset is complete, you will see “Master Reset Completed”, you need to reset or power down the Eggtimer before you can use it.

Memory Dump

The Memory Dump writes comma-separated-variable (CSV) data to the serial port containing the data in every memory location. This function is primarily used by the Eggtimer Technical

Support staff to debug problems that you may encounter, but you can also use it to save all of the flight data prior to a Master Reset if you choose.

To dump the memory, hold the Button down while the power is off or the Eggtimer is in reset mode, and continue to hold it down as you turn on the Eggtimer or release the reset. After about 10 seconds you will hear a long beep, acknowledging the Monitory mode, and you will see a "MON>" prompt on the screen. If you hit the "0" (zero) key, the last flight's memory will be dumped to the serial port; if you hit the "A" or "a" key, ALL flight memory will be dumped.

The memory dump format is:

Page No., Page Address, Memory Address, High Byte, Low Byte, Integer Value

Page No. : Page Number (0-31)
Page Address: Address within page (0-2047)
Memory Address: Address within memory (0-65534, by two's)
High Byte: Byte value of first byte
Low Byte: Byte value of second byte
Integer value: Value of integer (two's complement if negative)

Servo Set/Reset

If you have set the deployment channel(s) to servo mode, you can test the servo position and reset the servos to the opposite position from the Monitor mode. Note that BOTH servos will be set or reset at the same time, and this is only valid if the deployment channels have been previously set to servo mode (either 2 or 3).

To SET the servos (simulate deployment), enter the monitor mode by holding the Button down while turning on the Eggtimer. After about 10 seconds you will hear a long beep, and you will see a MON> prompt on the terminal screen. Hit the "S" key, and it both servos will be put into SET position.

To RESET the servos (rotate opposite of deployment), enter the monitor mode by holding the Button down while turning on the Eggtimer. After about 10 seconds you will hear a long beep, and you will see a MON> prompt on the terminal screen. Hit the "R" key, and it both servos will be put into RESET position.

Appendix B – Deployment Channel Testing

You can perform a pyro deployment test on your igniter channels using the Monitor mode. This allows you to make sure that the battery and the igniter that you are going to be using for deployment and/or airstarts are compatible, before actually flying them.

SAFETY WARNING: ALWAYS TEST IGNITERS OUTDOORS, WITH A SUITABLE DISTANCE BETWEEN YOU AND THE IGNITER (AT LEAST 5 FEET). If the igniter has a lot of pyrogen, i.e. it's going to be used for Airstarts, we strongly recommend that you put it in a tin can or other reasonably fire-safe container when you test it.

Note that only one deployment channel can be tested at a time, and the igniter will be fired for the amount of time that you set on the programming screen. This includes the "0 – continuous" setting, so make sure that you set it up properly ahead of time.

To test the CHA (Main) deployment channel, enter the monitor mode by holding the Button down while turning on the Eggtimer. After about 10 seconds you will hear a long beep, and you will see a MON> prompt on the terminal screen. Hit the "M" key, and you will hear a continuous warning beep. Press the Button, and the Main igniter channel will fire for the programmed amount of time

To test the CHB "B" deployment channel, enter the monitor mode by holding the Button down while turning on the Eggtimer. After about 10 seconds you will hear a long beep, and you will see a MON> prompt on the terminal screen. Hit the "B" key, and you will hear a continuous warning beep. Press the Button, and the Channel B igniter channel will fire for the programmed amount of time

To test the AUX output channels, enter the monitor mode by holding the Button down while turning on the Eggtimer. After about 10 seconds you will hear a long beep, and you will see a MON> prompt on the terminal screen. Hit the "1", "2", or "3" key depending on which AUX channel you want to test, and you will hear a continuous warning beep. Press the Button, and the selected AUX output will be turned on for two seconds.

Appendix C - Sample Flight Settings

Low-to-Medium Power Non-Deployment Example

These settings will work fine for a small D-size rocket, or most E-G rockets. If the rocket is heavy, you might want to take the Burn/Coast sample rate down a notch, to 5.

```
Eggtimer Build_1.44              Flight Settings
                                                                T:23.50
    A    100   LDA <50-500 by 50>                               P:97804.00
    B    8     Burn/Coast Samp/sec:  <4,5,8,10,15,20,25,33>     A:975.39
         125   Interval (ms)                                    Main:0
    C    2     Descent Samples/sec <1,2,4,5,10>                 CHB:0
         500   Interval (ms)                                    Batt:3.12
    D    0     Main Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW>
    E    300   Main Altitude <100-2000 by 50, 0 @ Apogee>
    F    2     Main ON-Time <1-9,0-Cont.>
    G    0     CHB Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW, 4-Airstart>
    H    2     CHB ON-Time <1-9,0-Cont.>
    I    1     Burn Timer <1-20>
    J    0     Airstart Delay <0-9>
    K    0     Telemetry


    O          Comments:    MPR non-deployment example



                   Options: Esc, A-O, ?
```

Low-Medium Power Multiple Deployment Example (Using Motor's Ejection)

If you add multiple deployment to the example above, you probably don't need to change the sample settings, but you obviously will need to set the deployment options. This example deploys the main chute at 300 feet, and fires for 2 seconds. The motor's ejection fires the drogue, so you don't need to set Channel B.

```
Eggtimer Build_1.44              Flight Settings
                                                                T:23.50
    A    100   LDA <50-500 by 50>                               P:97804.00
    B    8     Burn/Coast Samp/sec:  <4,5,8,10,15,20,25,33>     A:975.39
         125   Interval (ms)                                    Main:0
    C    2     Descent Samples/sec <1,2,4,5,10>                 CHB:0
         500   Interval (ms)                                    Batt:3.12
    D    1     Main Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW>
    E    300   Main Altitude <100-2000 by 50, 0 @ Apogee>
    F    2     Main ON-Time <1-9,0-Cont.>
    G    0     CHB Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW, 4-Airstart>
    H    2     CHB ON-Time <1-9,0-Cont.>
    I    2     Burn Timer <1-20>
    J    0     Airstart Delay <0-9>
    K    0     Telemetry


    O          Comments:    MPR multiple deployment



                   Options: Esc, A-O, ?
```

Higher Power Multiple Deployment (Using motor's ejection)

This example is for a much higher powered rocket, using the motor's ejection charge to fire the drogue near apogee. Since the projected altitude is likely to be higher than a lower powered rocket, you will most likely want to have the second deployment occur at about 800', higher than what you had set it for the mid-power example.

```
Eggtimer Build_1.44              Flight Settings
                                                                T:23.50
    A    200   LDA <50-500 by 50>                               P:97804.00
    B    20    Burn/Coast Samp/sec:  <4,5,8,10,15,20,25,33>     A:975.39
         50    Interval (ms)                                    Main:0
    C    2     Descent Samples/sec <1,2,4,5,10>                 CHB:0
         500   Interval (ms)                                    Batt:3.12
    D    1     Main Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW>
    E    800   Main Altitude <100-2000 by 50, 0 @ Apogee>
    F    2     Main ON-Time <1-9,0-Cont.>
    G    0     CHB Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW, 4-Airstart>
    H    2     CHB ON-Time <1-9,0-Cont.>
    I    3     Burn Timer <1-20>
    J    0     Airstart Delay <0-9>
    K    0     Telemetry


    O          Comments:    HPR w/motor ejection



                   Options: Esc, A-O, ?
```

High-Power Full Multiple Deployment Example

This example is for a larger rocket using multiple deployment, with both channels of the Eggtimer being used to fire charges. Other than setting Channel B to fire at Nose-Over, it's the same as the previous profile.

```
Eggtimer Build_1.44              Flight Settings
                                                                T:23.50
    A    200   LDA <50-500 by 50>                               P:97804.00
    B    20    Burn/Coast Samp/sec:  <4,5,8,10,15,20,25,33>     A:975.39
         50    Interval (ms)                                    Main:0
    C    2     Descent Samples/sec <1,2,4,5,10>                 CHB:0
         500   Interval (ms)                                    Batt:3.12
    D    1     Main Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW>
    E    800   Main Altitude <100-2000 by 50, 0 @ Apogee>
    F    2     Main ON-Time <1-9,0-Cont.>
    G    1     CHB Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW, 4-Airstart>
    H    2     CHB ON-Time <1-9,0-Cont.>
    I    3     Burn Timer <1-20>
    J    0     Airstart Delay <0-9>
    K    0     Telemetry


    O          Comments:    HPR multiple deployment


              Options: Esc, A-O, ?
```

Water Rocket Servo Deployment Example

These settings will work for a 2 litre or larger water rocket using a micro servo to trigger a parachute after apogee. Water rocketry is an art all to itself, if you haven't looked at it lately do a little Google'ing… you'll be amazed what they're doing. The Eggtimer is the only flight computer that will modulate servos without any extra electronics. All you need is a 39 ohm ¼ watt resistor across the deployment channel outputs to provide continuity and pull the servo's control input to ground when the channel isn't firing.

You will want a very low LDA, 50' should work. 10 samples per second should be appropriate, and 2 samples/sec for descent. We set the main channel to deploy at nose-over, with the servo's PWM at 6; this will give us pulse widths of 800ms and 2400ms lock-to-lock, which should make almost any servo go full range. The rocket is obviously not going anywhere near mach speeds, so we'll set the Burn Timer to the minimum value, 1.

```
Eggtimer Build_1.44             Flight Settings
                                                          T:23.50
    A    50    LDA <50-500 by 50>                         P:97804.00
    B    10    Burn/Coast Samp/sec:  <4,5,8,10,15,20,25,33>     A:975.39
         100   Interval (ms)                              Main:0
    C    2     Descent Samples/sec <1,2,4,5,10>           CHB:0
         500   Interval (ms)                              Batt:3.12
    D    2     Main Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW>
    E    0     Main Altitude <100-2000 by 50, 0 @ Apogee>
    F    6     Servo PWM <0-9>
    G    0     CHB Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW, 4-Airstart>
    H    2     CHB ON-Time <1-9,0-Cont.>
    I    1     Burn Timer <1-20>
    J    0     Airstart Delay <0-9>
    K    0     Telemetry


    O          Comments:    H20 rocket servo


              Options: Esc, A-O, ?
```

Appendix D - Setting Up Your Eggtimer for Airstarts

Channel B can be used to fire airstart igniters, however with airstarts you have to take extra safety precautions due to the consequences of an motor being accidentally fired on the pad. The Eggtimer requires three separate events, in order, for the igniter to be fired in Airstart mode:

1) LDA must have been reached

2) The Burn Timer and Airstart Delay timer, which are started by reaching LDA, must have expired.

3) The Breakwire must have been tripped. This actually happens on the pad right after liftoff, but it doesn't get checked until the Airstart Timer expires. If it's accidentally tripped prior to the "Flight Started" warbling tone, the Eggtimer will fail the self-test and the flight program will not start.

To pick the right settings for LDA, the Burn Timer, and the Airstart Delay, you must use a flight modeling program such as RocSim or OpenRocket, using the .rkt file for you model or as accurate a build as you can make. Check the CG and the weight carefully against your simulation. What you are looking for is the LDA and first stage burnout time and altitude. You will use these to figure out the LDA, Burn Timer, and Airstart Timer as follows:

1) Find the time that your rocket will hit LDA. We recommend 200' unless your rocket has a thrust-weight ratio of under 10, in which case 100' would be better.

2) Take the burn time of your motor and round it appropriately. This will be your value for the Burn Timer.

3) Add the Burn Timer value to the LDA time to the Burn Timer value and note the elapsed flight time from motor ignition. This will probably be slightly longer than your motor's burn time.

4) Determine where you want the second stage ignition to be. You may have to play around with this time in your modeling program to get the altitude/velocity that you want. If you want it to be at burnout, it's going to be the same as the value in #3, and you can simply set the Airstart Delay to zero.

5) Go backwards from the elapsed flight time of the second stage ignition to the expiration of the Burn Timer that you noted in #3, the difference will be the value of the Airstart Delay. Round up or down as appropriate to get the most accurate value. Note that if your igniter takes a significant amount of time to fire the motor (over 1 second) then you will need to SUBTRACT that time from the Airstart Delay, in order to have the second stage motor fire when you want it to.

You also need to ground test whatever igniter and battery combination you're going to use to make sure they will work together, you can use the CHB igniter test for this. (See Appendix B)

It is STRONGLY recommended that you use a good quality switch for the deployment battery, and that you shunt the igniter until immediately prior to launch, releasing it right before you start the Eggtimer's flight sequence by releasing the Reset switch. The sequence at the pad should be:

1) Connect your booster igniter
2) Turn on the Eggtimer, with the Reset switch enabled
3) Turn on the Deployment Battery
4) Release the Igniter Shunt
5) And approx. 1 minute before the flight, release the Reset switch
6) Assuming that self-test passes and you get the Flight Ready tone, you're ready to launch

If you get an error and you cannot start the flight, the power-down sequence should be:

1) Turn on the Reset Switch to put the Eggtimer on hold
2) Turn off the deployment battery
3) Reconnect the Igniter Shunt
4) Turn off the Eggtimer battery
5) Disconnect the booster igniter

The igniter is now shunted and the batteries are turned off, so it's impossible for the igniter to be fired. You may now correct the failed condition, and start over again.

Breakwire Tips

There are several ways that you can implement a breakwire, depending on how big your rocket is. For a low to medium power rocket, you can use a piece of aluminum foil across two small screws, and have a string tied to the pad that loops around the middle of the foil. When the rocket lifts off, the foil will tear off, opening the circuit. For a higher power rocket, you can use a piece of fine wire, similarly wrapped between two screws and broken by a heavy wire or Kevlar® line tied to the pad. The important thing is that whatever you are using for a breakwire can't break or come loose so easily that it might accidentally be broken while the rocket is simply sitting on the pad, or at the prep table.

You can also use some kind of microswitch with a lever that gets pushed back by the rail or rod, so that it releases when the rocket leaves the launcher. This has the advantage of being easy to prep because you don't have restring a wire across the terminals, and you don't have to install a tripwire on the pad. The disadvantage is that you need to be very careful to make sure that the switch closes completely while on the rail/rod, and that it can't accidentally be released until the rocket leaves the pad.

High-Power Airstart Programming Example

This example is for a rocket using Channel B in Airstart mode, with both channels of the Eggtimer being used to fire charges. In this case, the second stage motor's built-in ejection charge will be used to fire the drogue, so you will need to model the flight carefully in order to select the correct delay. The first stage's built-in ejection charge should be used to fire a chute to bring that stage down, it should be timed to pop right around the first stage's projected apogee. You may have to model the first stage's flight separately from the second stage to determine this value, but it will usually be a few seconds after second stage ignition because the added drag will slow the first stage rather quickly.

This example is for a rocket with a Thrust/Weight ratio of about 20/1, so the default 200' LDA setting will work fine; in fact, we don't really want to increase it, because that will delay the Burn Timer's start.

We are assuming that the main motor will fire for approximately two seconds, and that we want a 1 second delay after burnout before firing the second stage. If we know that our igniter will take about one second to ignite the second stage motor, the actual delay from burnout will be the one second, which happens to be the delay that we wanted. Therefore, we will set the Airstart Delay to zero which will fire the igniter as soon as the Burn Timer ends, with the second stage motor igniting about one second afterwards. We also increased the CHB On-Time to 3 seconds (from 2), since the igniter may take longer to heat up than a simple electric match used to fire ejection charges. We recommend that you test the igniter-motor combination on a single-stage flight first or consult other club members, forums, etc., so you have an idea of how long it takes that particular igniter to fire that particular motor.

```
Eggtimer Build_1.44             Flight Settings
                                                                T:23.50
    A    200   LDA <50-500 by 50>                               P:97804.00
    B    20    Burn/Coast Samp/sec:  <4,5,8,10,15,20,25,33>     A:975.39
         50    Interval (ms)                                    Main:0
    C    1     Descent Samples/sec <1,2,4,5,10>                 CHB:0
         1000  Interval (ms)                                    Batt:3.12
    D    1     Main Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW>
    E    500   Main Altitude <100-2000 by 50, 0 @ Apogee>
    F    2     Main ON-Time <1-9, 0-Cont.>
    G    4     CHB Enable <0-Off, 1-IGN, 2-Servo CCW, 3-Servo CW, 4-Airstart>
    H    3     CHB ON-Time <1-9,0-Cont.>
    I    2     Burn Timer <1-20>
    J    0     Airstart Delay <0-9>
    K    0     Telemetry


    O          Comments:    Airstart example

              Options: Esc, A-O, ?
```

Appendix E - Specifications

Pressure Sensor:	
Type	Bosch BMP-85/180, accuracy rated to approx. 9000 m (29,500 ft)
Accuracy	Approx 1m (3 ft) throughout rated pressure range
Precision	Approx ½ m (1.5 ft) throughout rated pressure range
Processor	
Type	Atmel ATMega328P
Speed	16 MHz
Memory	
Type	I2C Serial EEPROM
Size	64KB (512 Kb)
Deployment	
Voltage	3v-60v
Current	8A max, 60 W
Isolation	> 3000V (opto-isolator)
Duration	1-9 seconds, or continuous
Output	On, or pulse-width-modulated output for analog servos
Servo PWM	0.5-2.5 ms (selectable +/- 100 ms pulse width increments)
AUX Channels (3)	3v-4v at 20 mA max (logic-level outputs)
Size	Approx 3.6” L x 1.1” W x .5”H
Weight	Approx 20 grams
Power	
Voltage	3-4v nominal
Current Draw	
Reset	Approx 2 ma
Operational	Approx 8 ma