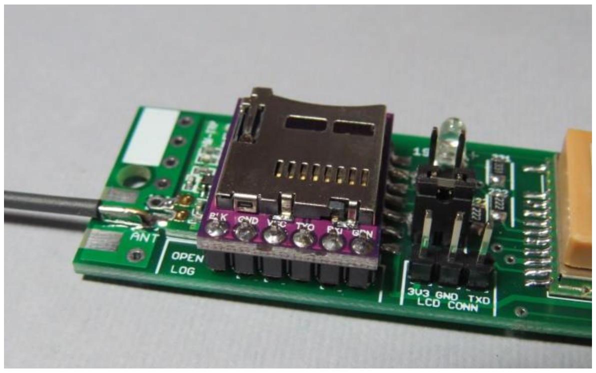
Eggfinder and OpenLog Users Manual

All Eggfinder transmitters, and the Eggfinder TRS as well, can be connected to a standard OpenLog datalogger to record the NMEA data track. The OpenLog is special Arduino-based board what is designed to simply read serial data streams and write them to a file on a micro SD card (uSD from now on). They are available from many vendors for well under $20, including Eggtimer Rocketry.

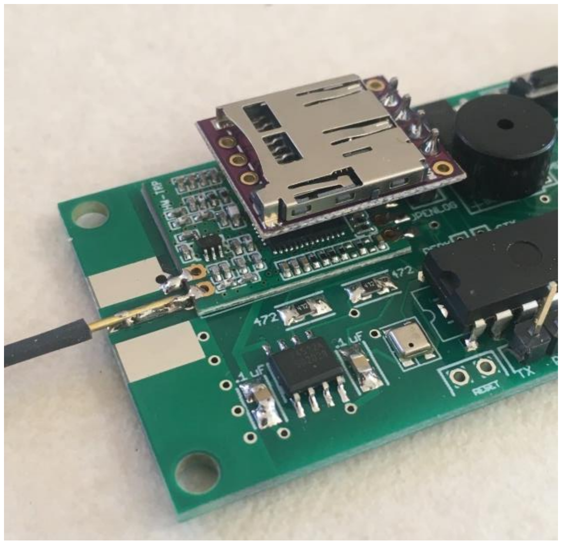
Adding an OpenLog module to your Eggfinder TX, Eggfinder Mini, or Eggtimer TRS gives you a new level of data that you can play with. With some simple and free tools such as Google Earth, you can take the NMEA GPS data from your Eggfinder and create a 3D graph of your flight. This is particularly helpful for analyzing upper level winds, or for Level 3 flights or other flights in which you need to show that your rocket landed "in bounds".

Adding an OpenLog Transmitter to Your Eggfinder TX or Eggtimer TRS

If you have an Eggfinder TX Rev. C1 or an Eggtimer TRS Rev. C1, adding an OpenLog is very easy, there's already a place for the header. All you need to do is to solder the header to the board, and solder the other side of the header to the OpenLog module… instructions are in the Eggfinder TX Assembly Guide.



Eggfinder TX RevC1 w/OpenLog



Eggtimer TRS RevB4 w/OpenLog

If you have an Eggfinder TX Rev. B6 or B7, there's a special adapter board that simply slides over the existing headers.   You simply tack solder it in place and you're good to go.

If you have an older Eggtimer TRS, or an older Eggfinder TX that doesn't support the adapter board, you can still connect an OpenLog, but you'll have to run a few wires…three to be exact.   You're going to connect them directly to the pads on the RF module, using #24 or smaller wire. Connect the OpenLog to the RF Module as follows:

OpenLog VCC Pad ----- FIRST pad on the left on the RF Module
Open Log GND Pad ----   THIRD pad on the left on the RF Module
OpenLog RXD Pad ---- FOURTH pad on the left on the RF Module

Preparing your uSD Card

Before using your uSD card in the OpenLog, we recommend that you format it.   The OpenLog firmware expects the file format to be FAT32, some of the cards that we have seen (particularly those for cameras) come formatted with another format.   You'll need a uSD-SD card adapter to do this on your Windows PC (you'll need one anyway, to read the data).

Starting up the OpenLog

When you put a blank uSD card in the OpenLog and power up your Eggfinder, it should start logging data immediately. You'll be able to tell, because the blue (or sometimes yellow) LED on the OpenLog will start blinking in sync with the red LED on the RF module.   This means that it's receiving data from the transmitter, you'll also see the green LED on the OpenLog blink afterwards too, that means that it's writing data to the uSD card.   Let it run for 10 or 15 seconds, then power it off. Remove the uSD card and put in in the SD card adapter, then insert it in your PC. If you open up the card, you should see two files…

· config.txt - This is the config file that the OpenLog uses to set baud rate, etc.
· lognnnnn.txt – This is the data log, starting from "00000" and going up to "99999".

The config.txt File

The config.txt file is read by the OpenLog at startup, and tells it how to configure such things as baud rate, commands, etc. If the uSD card is blank or there is no config.txt file, one is created with defaults. In most cases, the default is fine… it supports 9600 baud data feeds, which is what the Eggfinder sends out.

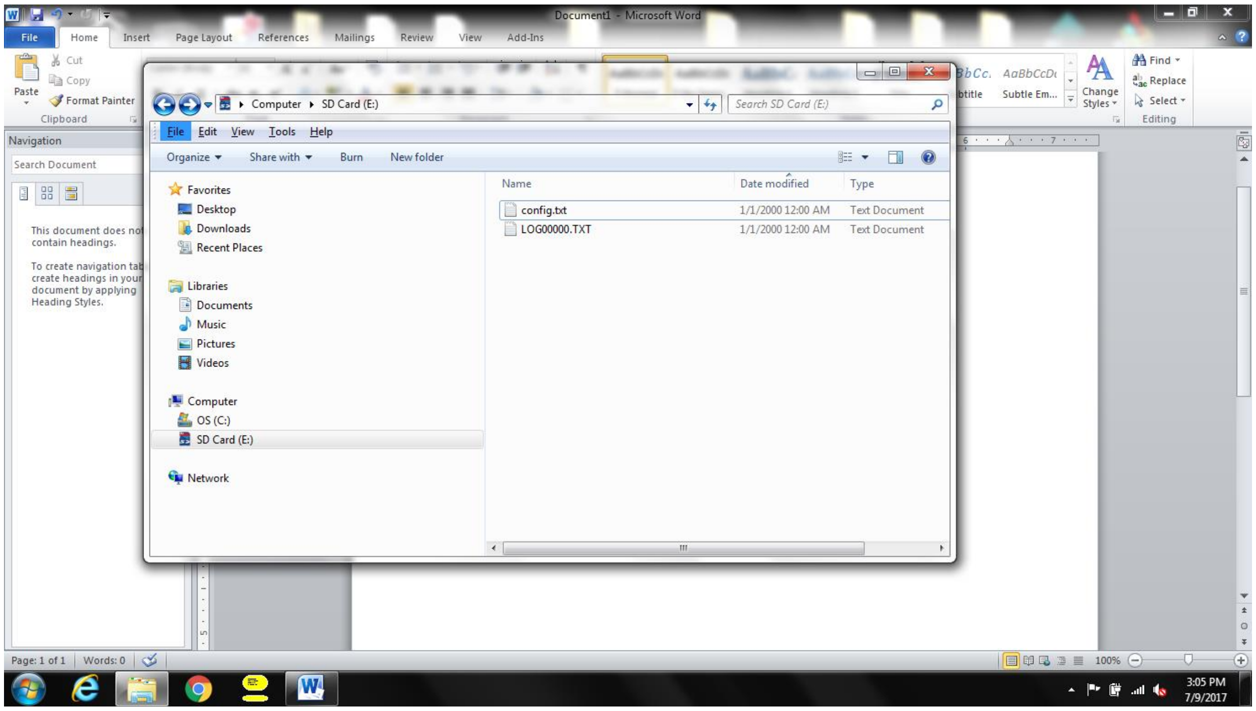
Some OpenLog versions default to 115,200 baud, which is obviously not correct.   You can tell if this is the case because the log000000.txt file will be a bunch of random characters instead of readable text. If this is the case, you can replace the file with the appropriate file, which you can download from the Eggtimer Rocketry web site under Eggfinder Support/Downloads. Since it's a very small file, here are the contents:

```
9600,26,3,0,0,0,0
baud,escape,esc#,mode,verb,echo,ignoreRX
```

The “9600” is the baud rate, “26” is the escape character that puts it into command mode (ctrl-z) and is not used by the Eggfinder, the “3” is the number of escape characters that must be received in succession to enter the command mode, and the zeroes are option characters that all must be zero for it to work with the Eggfinder. If your config.txt file doesn’t look like this, simply edit it so that it does. We also recommend that you make it read-only, so it doesn’t get overwritten; we’ve seen it happen, and the resulting baud rate mismatch causes the data to be garbled and useless.

The lognnnnnn.txt Files

The lognnnnnn.txt files are the data files that are create by the OpenLog. It is a simple dump of whatever characters are fed into it, in the case of the Eggfinders it's a NMEA-formatted GPS data stream. They are typically named "LOG" + five digits + ".txt"; for example, the first one that is written to the uSD card may be named "LOG00000.txt" although sometimes they start with 256 so you'll see a file "LOG00256.txt" as the first file.



Name	Date modified	Type
config.txt	1/1/2000 12:00 AM	Text Document
LOG00000.TXT	1/1/2000 12:00 AM	Text Document

Note that the extension of this file is always ".txt". The OpenLog has no clue what data you're feeding to it… it simply logs whatever it gets. In some applications this would be a disadvantage because you might want to do some real-time processing, but for post-flight processing it's great to have whatever data gets fed to it unaltered. Particularly in the case of the TRS, this allows you to get a time-stamp of the altitude and deployment channel status records that get sent to the LCD receiver.

Graphing the NMEA Data Records with Google Earth

One of the most common programs for creating a 3D flight profile is Google Earth. Google Earth has the capability of reading many different kinds of GPS data streams and graphing them; NMEA data is the most basic data stream since it's the native output of almost all GPS modules, so obviously that capability is common.

Extracting the Altitude an Status Data from the Eggtimer TRS

Along with the standard NMEA GPS data stream, the Eggtimer TRS sends altitude and deployment status records. They are sent once per second, although the TRS samples the altitude 20 times per second until nose-over and two samples per second afterwards.

The format of these records is:

Altitude: <altitude>
Status: {dm} where "d" is the Drogue channel status and "m" is the Main channel status

The Altitude records are pretty straight-forward… if you see a record like <10012> that means that your Above Ground Level (AGL) altitude is 10012 at that point. If you compare those records to the GPS data records you can clearly see the lag in GPS altitude reporting, which can be really far off during ascent but usually gets much better as the rocket descends under chute.

The Status records are a little different, because each channel can have 3 values:

- D or M: Channel has continuity AND has not been fired.
- d or m: Channel has continuity AFTER firing.

(i.e. bridgewire didn't melt, or you have some other kind of deployment mechanism such as a solenoid)

- spaces: Channel does not have continuity OR was not selected.

So for example, if you get something like { M} that means that the drogue doesn't have continuity or was not selected, and the Main has continuity but has not been fired. If you get something like {dM} that means that the Drogue was fired but still shows continuity, and the Main has continuity but has not been fired. If you get { } that means that either both channels do not have continuity or neither one of them was selected.