# DIY-ish Plane Altitude / Temperature / GPS logger



## Components

* Ardulog – arduino based serial to SD card data logger
* BMP085 pressure / temperature sensor
* Custom adapter board
* Custom firmware for Ardulog to read BMP085
* Windows software for viewing log file

## Description

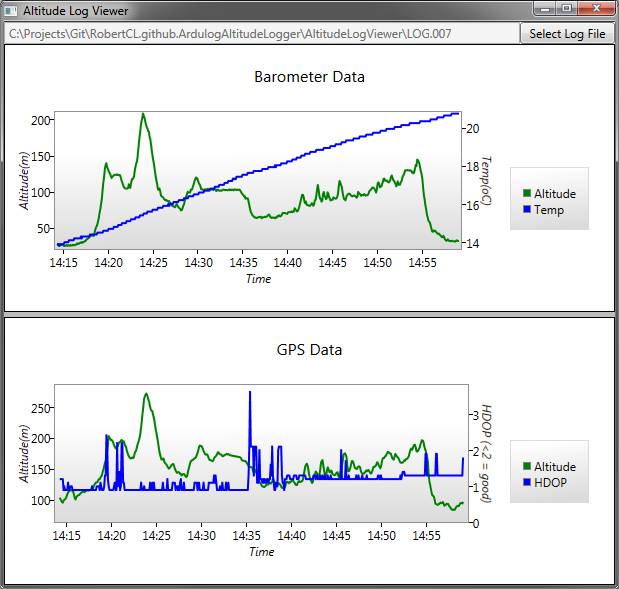
SD card flight logger. Logs pressure, temperature, altitude (calculated from pressure) to the card once per second and GPS data (if connected) whenever the GPS reports it (also once per second).

## Logger Usage

* Setup Config.txt on micro SD card
  + BAUD: 34800
    - Baud rate for serial connection (don’t change)
  + FILE: LOG
    - Prefix for log file name (don’t need to change)
  + PASL: 0
    - Initial value for Pressure At Sea Level – used to calculate altitude. Leave as zero to take an initial reading at start-up, then 0m altitude in the logs will be ground level
* (optional) Plug in GPS
  + TXO (TX Out) -> RXI (RX In)
  + GND -> GND
* Insert SD card
* Connect 5-6V to the power wires (black and red)
* Fly! (trying not to crash)

## Windows Software

* Press Select Log File (or drag and drop a file onto the button)



Two graphs will be drawn. The first shows the BMP085 data - calculated altitude (green) and temperature (blue). The second shows some basic data from the GPS – altitude and HDOP (an indication of the quality of the GPS fix).

Actual time will be calculated for the x-axis if GPS data is present, otherwise it will show time from the start of the flight.

There will also now be two additional files created alongside the original log file – one containing just the barometer data and one containing just the GPS data.

The GPS data file can be loaded into (eg) <http://www.gpsvisualizer.com/> to produce a flight map on Google maps or Google Earth kml file to be viewed in Google earth. The GPS data file is in standard NMEA format (just the raw data that the GPS outputs) so should be useable in any software that will read GPS data.

## File Formats

Two types of lines get logged to this file starting with either “BMP085” or “GPS”.

BMP085,3000,139,101016,25.76

* BMP085 – line identifier
* 3000 – elapsed milliseconds since start-up
* 139 – temperature x 10. So 139 = 13.9oC
* 101016 – raw pressure in Pa
* 25.76 – calculated altitude in meters

GPS,4000,$GPVTG,13.83,T,,M,0.00,N,0.0,K,A\*34  
GPS,4000,$GPGGA,141420.000,5131.0372,N,00152.8219,W,1,08,1.2,101.2,M,48.1,M,,0000\*4C  
GPS,4000,$GPGSA,A,3,14,11,32,28,20,01,17,31,,,,,1.6,1.2,1.2\*3C  
GPS,4000,$GPGSV,3,1,10,01,75,107,33,32,73,074,26,20,66,238,34,11,48,135,31\*78  
GPS,4000,$GPGSV,3,2,10,17,39,302,31,23,19,179,,14,17,038,25,28,13,251,26\*75  
GPS,4000,$GPGSV,3,3,10,19,08,159,13,31,08,085,21\*73  
GPS,4000,$GPRMC,141420.000,A,5131.0372,N,00152.8219,W,0.00,13.83,171113,,,A\*47

* GPS – line identifier
* 4000 – elapsed milliseconds since start-up
* $GPxxx standard GPS NMEA sentences received from GPS

## On The CD

* \Arudlog – custom firmware for Ardulog board, including code to read the BMP085 pressure sensor. Open ardulog.ino in the Arduino IDE to edit / compile – you’ll need a USB FTDI cable if you want to actually program the Ardulog board.
* \AltitudeLogViewer – Windows software, C#
* Config.txt – sample config file for SD card

## TODO

* Windows software is a bit buggy and doesn’t like badly formatted log files
* Altitude calculation relies on initial pressure reading, which is sometimes off. Could do with some kind of averaging, the BMP085 code appears to have a setting for configuring Oversampling, which might do the trick.
* Sometimes some rows in the log file get some corruption
* …

## Links

* BMP085 tutorial - <https://www.sparkfun.com/tutorials/253>
* Ardulog - <http://www.hobbytronics.co.uk/ardulog2>