

# Built In Logging

The built in logging capability isn't easy to use with the Shield - we wanted to make it better for SD logging so while we do have some details here, its not well support and not recommended

For more details, you can read the [LOCUS \(built-in-datalogging system\) user guide](#)

# Built In Logging

One of the nice things about the GPS module on the shield is the built in data-logger. This basic data-logging capability can store date, time, latitude, longitude and altitude data into a 64K flash chip inside. Its not a high resolution logger - it only logs once every 15 seconds when there is a fix - but for some projects that want to track location, this can be a great low power way to log data - no SD card required! It can store up to ~16 hours of data.

The GPS module does require the Arduino to 'kick start' the logger by requesting it to start. If power is lost it will require another 'kick' to start. If you already have some data in the FLASH, a new trace will be created (so you wont lose old data) and if you run out of space it will simply halt and not overwrite old data. Despite this annoyance, its still a very nice extra and we have some library support to help you use it

First, we should try getting the logger to run. Make sure the switch is set to SoftSerial

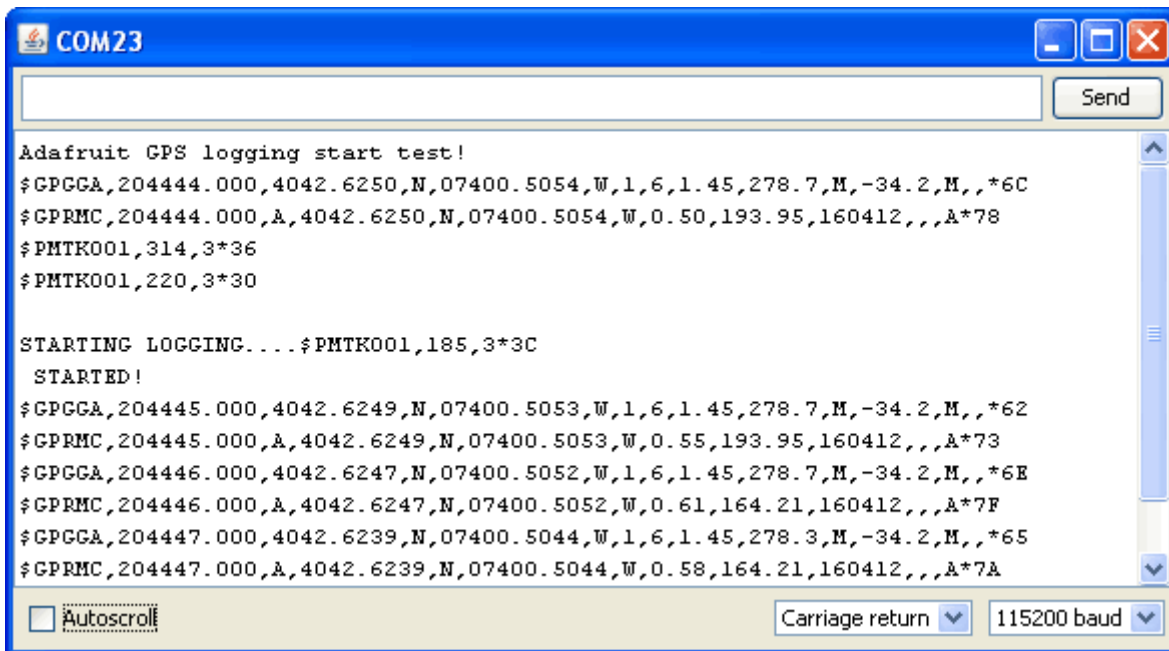
Open up the **File**→**Examples**→**Adafruit\_GPS**→**locus\_start** sketch and change the line

```
SoftwareSerial mySerial(3, 2);
```

to

```
SoftwareSerial mySerial(8, 7);
```

and upload it to the Arduino. Then open up the serial monitor. If you have an Uno or compatible you will see the echo'd data. Leonardos will not see the echo data but you should still see a message that says "STARTING LOGGIN...STARTED!"



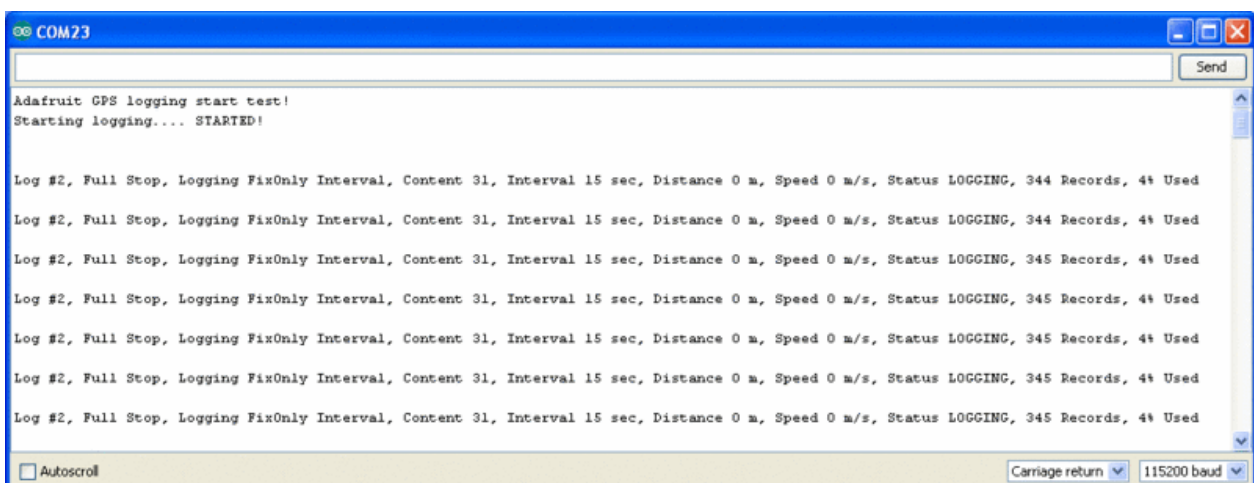
## Logging Status

Once you've seen that the GPS is OK with logging, you can load up the status sketch which will also give you more data. Upload **File**→**Examples**→**Adafruit\_GPS**→**locus\_status** and change the line

**SoftwareSerial mySerial(3, 2);**

to

**SoftwareSerial mySerial(8, 7);**



# Downloading Data

Finally, once we're done logging we need to extract the data. To do this we need to first get the raw data out of the FLASH and then decode the sentences.

UploadFile→Examples→Adafruit\_GPS→locus\_dump to the Arduino and open up the serial monitor. Change the software serial line to use pins 8 and 7

PLEASE NOTE: Asking the Arduino, with 2K RAM buffer to handle 64KB of FLASH data and spit it out from the GPS can sometimes over-tax the processor. If you are having hiccups, check the GPS tool instructions below Copy and paste all the text after the ---'s (starting with \$PMTKLOX,0,86\*67 and ending with \$PMTK001,622,3\*36) [then paste it into the box located on this page.](#)

## Using the GPS Tool

If you are having difficulty with the Arduino/javascript tool, you can also try using the GPS tool. The tool runs only under Windows but it is very powerful.

You can only do this with an Uno/Duemilanove/Diecimila/Mega!

You'll also need to set up direct wiring. Go back to the [connect with Direct Wiring example](#), and get that working. You'll FTDI adapter or other TTL converter and download the [GPS Tool](#) - connect to the GPS via the COM port of the Arduino/FTDI/TTL cable. You can then query, dump and delete the log memory