

gpsLogger 1.8

gpsLogger is a simple program which monitors a serial port for data from a Global Positioning System (GPS) receiver (NMEA format). *gpsLogger* "publishes" the current position to shared memory where other client applications can "subscribe" to read the current GPS position. A publish/subscribe application program interface API is defined in the "gpsPub.h" file in the *gpsLogger* source code distribution. The NRL Multi-Generator (MGEN) test tool set uses this methodology for embedding GPS position information in messages it generates. This tool set has been used for mobile wireless network performance evaluation and is available from <https://www.nrl.navy.mil/itd/ncs/products/mgen>. By default, *gpsLogger* will also log position information to `stdout` in the comma-delimited lines of the format:

```
time><hr>:<min>:<sec> position><longitude>,<latitude>,<altitude>
```

The time logged is the computer system time in Greenwich Mean Time (GMT). The <latitude> and <longitude> are given in units of degrees. Positive <latitude> indicates North, negative <latitude> indicates South, positive <longitude> indicates East, negative <latitude> indicates West, and <altitude> is in units of meters.

Note that *gpsLogger* can optionally set the system time and synchronize it to GPS. When *gpsLogger* is used to set the system time, it can work in one of two ways:

- 1) If the `pps` option is **not** specified, the system time is set once (and only once) upon the first valid timestamp received from the GPS device. This time set is accurate to within approximately +/- one second, with subsequent system clock drift eventually diverging the system time from GPS time.
- 2) If the `pps` option is specified, *gpsLogger* will accurately (often within 1 msec) set and adjust the system time to track GPS time. The `adjtime()` system call is used to keep the system time in sync with GPS while preserving a monotonically increasing system clock time. However, if the system time diverges more than one second from GPS time, the `settimeofday()` call is used to force the system time in sync with GPS (which may result in a negative system time shift).

An explicit `force` command-line option is provided so that if the system time is already within one second of GPS time when *gpsLogger* is started, it will immediately "force" the system time into sync with GPS using the `settimeofday()` call on its first time adjustment instead of `adjtime()` which could possibly take a long period of time to converge towards GPS if the time delta is close to one second. The `force` option is provided for applications where

the user doesn't care about an initial negative system time shift and wants to quickly get system time in sync with GPS rather than waiting for a possibly lengthy convergence to GPS time. After the initial "forced" hard time adjustment, *gpsLogger* behaves normally, generally using the `adjtime()` call to keep the system time in sync with GPS, and only invoking `settimeofday()` when the system time strays more than one second from GPS time.

The *gpsLogger* utility was created as a light-weight alternative to the Network Time Protocol daemon (*ntpd*), also providing the GPS position information in shared memory for use by other applications or processes. The *ntpd* program offers much more flexibility when time synchronization is the only concern.

Download

The *gpsLogger* package is available at:

<https://github.com/USNavalResearchLaboratory/gpsLogger>

Files:

<code>gpsLogger.cpp</code>	gpsLogger source code
<code>nmeaParse.h</code> <code>nmeaParse.cpp</code>	Routines for parsing NMEA sentences
<code>gpsPub.h</code> <code>gpsPub.cpp</code>	Routines for GPS position publish/ subscribe (using shared memory)
<code>gpsClient.cpp</code>	Example C++ source code for using <code>GPSSubscribe()</code>
<code>gpsFaker.cpp</code>	Program to publish "fake" GPS position.

To Build

```
gcc -o gpsLogger gpsLogger.cpp nmeaParse.cpp gpsPub.cpp
```

Usage:

```
gpsLogger [set][pps][force][check][gps35][noLog]  
          [log <logFile>][device <serialDevice>]  
          [speed <baudRate>][debug]
```

Options:

<code>set</code>	Cause <i>gpsLogger</i> to set and synchronize the system time to GPS time. Should be used with the <u>pps</u> option for the best accuracy.
<code>pps</code>	Causes <i>gpsLogger</i> to look for a pulse-per-second (PPS) signal on the serial port data carrier detect

	(DCD) line to use as a reference for accurate sync of system time to GPS time.
force	Causes <i>gpsLogger</i> to hard sync the system time of day to GPS time using a call to "settimeofday()" on the first time read from the GPS device. This speeds up convergence of system time to GPS time, but may result in the system's time moving backwards on the first sync to GPS time.
check	Causes <i>gpsLogger</i> to <u>require</u> valid checksums on NMEA sentences from the GPS device.
gps35	Causes <i>gpsLogger</i> , at startup, to attempt to configure a Garmin GPS-35, GPS-16 (and possibly other Garmin or other models) for PPS operation. The <u>pps</u> option must also be used for PPS operation.
noLog	Cause <i>gpsLogger</i> to not output logged position information. The position information is only available via the published shared memory location.
log <logFile>	Causes <i>gpsLogger</i> to output position information as described above to the file named <logFile>. The <u>noLog</u> option overrides this behavior.
debug	Causes <i>gpsLogger</i> to output debugging information to stderr while running.
device <serialDevice>	This option allows the user to specify the serial port to be monitored. By default, the device "/dev/ttyS0" is monitored.
speed <baudRate>	This option determines the baud rate configuration of the serial port. Valid values include 4800 and 9600. By default, the <baudRate> is 4800.
debug	This option causes <i>gpsLogger</i> to output debugging information to stderr including NMEA sentences read from the GPS device, etc.

Notes:

- 1) For the gps35 option, *gpsLogger* uses the NMEA "PGRMC" configuration sentence. This may work with other GPS device manufacturers and models which support PPS operation. PPS operation may be available without use of this option if the device has already been configured for PPS operation.
- 2) When the pps option is invoked, the GPS <altitude> information may be unavailable or unreliable. This is because the serial port is flushed to properly read the NMEA sentence containing time information associated with the corresponding pulse. As a result, the NMEA

"GPGGA" sentences containing the <altitude> information may be missed.

- 3) With regards to setting and synchronizing the computer system time to GPS time, *gpsLogger* uses the `settimeofday()` system function to abruptly adjust the time when the time differential is greater than one second. Otherwise, the `adjtime()` system call is used to preserve a monotonically increasing system clock.