

- [Home](#)
- [Libraries](#)
 - [TinyGPS](#)
 - [NewSoftSerial](#)
 - [Streaming](#)
 - [PString](#)
 - [Flash](#)
 - [PWMServo](#)
- [About](#)
- [Projects](#)
 - [The Reverse Geocache™ Puzzle](#)
 - [Building a Puzzle Box](#)
 - [More Puzzle Box Tales](#)

Email: [Admin](#) [RSS](#) Subscribe: [RSS feed](#)

[Arduiniana](#)

Arduino wisdom and gems by Mikal Hart

TinyGPS

A Compact Arduino GPS/NMEA Parser

Note: TinyGPS 12 is now available. TinyGPS 12 provides several new features:

1. `satellites()` – Number of satellites counter from \$GPRMC
2. `hdop()` – Horizontal Dilution of Precision from \$GPRMC — the smaller this value is the more confidence you have in the accuracy of the location reading.
3. `course_to()` – another Maarten Lamers' borrowing that complements his `distance_to`
4. `cardinal()` – a nice little static function by Matt Monson that converts a course to a compass direction like "SSE".

Note: TinyGPS 11 is now Arduino 1.0 compatible.

TinyGPS is designed to provide most of the NMEA GPS functionality I imagine an Arduino user would want – position, date, time, altitude, speed and course – without the large size that seems to accompany similar bodies of code. To keep resource consumption low, the library avoids any floating point dependency and ignores all but a few key GPS fields.

Usage

To use, simply create an instance of an object like this:

```
#include "TinyGPS.h"
TinyGPS gps;
```

Feed the object serial NMEA data one character at a time using the **encode()** method. (TinyGPS does not handle retrieving serial data from a GPS unit.) When **encode()** returns "true", a valid sentence has just changed the TinyGPS object's internal state. For example:

```
#define RXPIN 3
#define TXPIN 2
SoftwareSerial nss(RXPIN, TXPIN);
void loop()
{
  while (nss.available())
  {
    int c = nss.read();
    if (gps.encode(c))
    {
      // process new gps info here
    }
  }
}
```

You can then query the object to get various tidbits of data. To test whether the data returned is stale, examine the (optional) parameter "fix_age" which returns the number of milliseconds since the data was encoded.

```
long lat, lon;
unsigned long fix_age, time, date, speed, course;
unsigned long chars;
unsigned short sentences, failed_checksum;

// retrieves +/- lat/long in 100000ths of a degree
gps.get_position(&lat, &lon, &fix_age);

// time in hhmmsscc, date in ddmmyy
gps.get_datetime(&date, &time, &fix_age);

// returns speed in 100ths of a knot
speed = gps.speed();

// course in 100ths of a degree
course = gps.course();
```

Statistics

The **stats** method provides a clue whether you are getting good data or not. It provides statistics that help with troubleshooting.

```
// statistics
gps.stats(&chars, &sentences, &failed_checksum);
```

- `chars`

- the number of characters fed to the object
- sentences
- the number of valid \$GPGGA and \$GPRMC sentences processed
- failed_checksum
- the number of sentences that failed the checksum test

Integral values

Values returned by the core TinyGPS methods are integral. Angular latitude and longitude measurements, for example, are provided in units of 10^{-5} degrees, so instead of 90°30'00", `get_position()` returns a longitude value of **9050000**, or 90.5 degrees. But...

Using Floating Point

...for applications which are not resource constrained, it may be more convenient to use floating-point numbers. For these, TinyGPS offers several **inline** functions that return more easily-managed data. Don't use these unless you can afford to link the floating-point libraries. Doing so may add 2000 or more bytes to the size of your application.

```
float flat, flon;

// returns +/- latitude/longitude in degrees
gps.f_get_position(&flat, &flon, &fix_age);
float falt = gps.f_altitude(); // +/- altitude in meters
float fc = gps.f_course(); // course in degrees
float fk = gps.f_speed_knots(); // speed in knots
float fmph = gps.f_speed_mph(); // speed in miles/hr
float fmps = gps.f_speed_mps(); // speed in m/sec
float fkmph = gps.f_speed_kmph(); // speed in km/hr
```

Date/time cracking

For more convenient access to date/time use this:

```
int year, month, day, hour, minutes, second, hundredths;
unsigned long fix_age;

gps.crack_datetime(&year, &month, &day,
    &hour, &minute, &second, &hundredths, &fix_age);
```

Establishing a fix

TinyGPS objects depend on an external source, i.e. its host program, to feed valid and up-to-date NMEA GPS data. This is the only way to make sure that TinyGPS's notion of the "fix" is current. Three things must happen to get valid position and time/date:

1. You must feed the object serial NMEA data.
2. The NMEA sentences must pass the checksum test.
3. The NMEA sentences must report valid data. If the \$GPRMC sentence reports a validity of "V" (void) instead of "A" (active), or if the \$GPGGA sentence reports fix type "0" (no fix) then those sentences are discarded.

To test whether the TinyGPS object contains valid fix data, pass the address of an unsigned long variable for the "fix_age" parameter in the methods that support it. If the returned value is **TinyGPS::GPS_INVALID_AGE**, then you know the object has never received a valid fix. If not, then fix_age is the number of milliseconds since the last valid fix. If you are "feeding" the object regularly, fix_age should probably never get much over 1000. If fix_age starts getting large, that may be a sign that you once had a fix, but have lost it.

```
float flat, flon;
unsigned long fix_age;

// returns +/- latitude/longitude in degrees
gps.f_get_position(&flat, &flon, &fix_age);
if (fix_age == TinyGPS::GPS_INVALID_AGE)
    Serial.println("No fix detected");
else if (fix_age > 5000)
    Serial.println("Warning: possible stale data!");
else
    Serial.println("Data is current.");
```

Interfacing with Serial GPS

To get valid and timely GPS fixes, you must provide a reliable NMEA sentence feed. If your NMEA data is coming from a serial GPS unit, connect it to Arduino's hardware serial port, or, if using a "soft" serial port, make sure that you are using a reliable SoftSerial library. As of this writing (Arduino 0013), the SoftwareSerial library provided with the IDE is inadequate. It's best to use my [NewSoftSerial library](#), which builds upon the fine work ladyada did with the [AFSoftSerial](#) library.

Library Version

You can retrieve the version of the TinyGPS library by calling the static member `library_version()`.

```
int ver = TinyGPS::library_version();
```

Resource Consumption

Linking the TinyGPS library to your application adds approximately 2500 bytes to its size, unless you are invoking any of the `f_*` methods. These require the floating point libraries, which might add another 600+ bytes.

Download

The latest version of TinyGPS is available here: [TinyGPS12.zip](#)

Change Log

1. initial version
2. << streaming, supports \$GPGGA for altitude, floating point inline functions
3. also extract lat/long/time from \$GPGGA for compatibility with devices with no \$GPRMC
4. bug fixes
5. API re-org, attach separate fix_age's to date/time and position.
6. Prefer encode() over operator<<. Encode() returns boolean indicating whether TinyGPS object has changed state.
7. Changed examples to use NewSoftSerial in lieu of AFSofSerial; rearranged the distribution package.
8. Greater precision in latitude and longitude. Angles measured in 10^{-5} degrees instead of 10^{-4} as previously. Some constants redefined.
9. Minor bug fix release: the fix_age parameter of get_datetime() was not being set correctly.
10. Added Maarten Lamers' distance_to() as a static function.
11. Arduino 1.0 compatibility
12. Added satellites(), hdop(), course_to(), and cardinal()

Acknowledgements

Many thanks to Arduino forum users [mem](#) and [Brad Burleson](#) for outstanding help in alpha testing this code. Thanks also to Maarten Lamers, who wrote the [wiring library](#) that originally gave me the idea of how to organize TinyGPS. Thanks also to Dan P. for suggesting that I increase the lat/long precision in version 8. Thanks to many people who suggested new useful features for TinyGPS, especially Matt Monson, who wrote some nice sample code to do so.

All input is appreciated.

Mikal Hart

Page last updated on February 8, 2012 at 5:02 pm

240 Responses → "TinyGPS"

[← Older Comments](#)

1.

Gatech

1 week ago

Hey mikal: I have tried some example codes for using my new Skylab SKM53 GPS Module with my Arduino Board.

I connected it well, but codes aren't working.

I got that gps.encode(GPS.read()) returns FALSE, and I never get and print a position.

I think my GPS Module is bad Sad, but I added a Serial.print(GPS.read()); before gps.encode(...) and get responses from the GPS like this (on Processing's Serial Monitor):

```
36807144495753535356544648487848484848444444449554877494644444252107171657749
44444444444508342576748534848486946484913
```

```
36807144495756535356544648487848484848444444449554877494644444257107171657749
44444444444508342574448444848486948554910
```

All the same length...

Is it broken?. What am I doing bad?.

Regards 😊

2.

[Mikal](#)

1 week ago

@bklw,

Not sure I understand the question, but if the time fix status is GPS_INVALID_FIX_TIME, then TinyGPS hasn't seen any valid time.

3.

[Mikal](#)

1 week ago

@ Kyle,

Congratulations—you finally pushed me over the edge and I'm soon posting TinyGPS 12, which supports satellites and HDOP as well as a couple of other goodies. 😊 Yay! I'm actually really happy to have those in there, and I intend to use them in my own projects. Thanks for the prod and for everyone who has asked for those.

4.

[Mikal](#)

1 week ago

@Gatech,

Hmm... Instead of Serial.print(), can you Serial.write() those values? I think you're getting real data, although not quite the type TinyGPS expects. Here's a partial decode using <http://www.asciitable.com>:

```
36 80 71 44 49 57 53 53 56 54 46 48 48 78 48 48 48 48 44 44 44 49 55 48 77 49 46 44 44 42 52 10 71 71 65 77 49
$PG,1955586.00N00000,,,
```

It looks a lot like NMEA data. Sort of.

Also, make sure you aren't losing characters. Make sure you are talking to the Serial port at 115200K Baud, so that you write as quickly as possible.

5.

UrSuS

1 week ago

@Mikal, @Kyle,

Thanks for reply guys!

Will wait for the new version 12 with goodies! Hooray! 🙌

6.

Gatech

1 week ago

Thanks. I'm getting:

```
$PMTK011,MTKGPS*08
$PMTK010,001*2E
$GPGGA,015540.220,8960.0000,N,00000.0000,E,0.0,,137.0,M,13.0,M,,*46
$GPGSA,M,1,,,,,,,,,,,,,*12
$GPGSV,1,1,00*79
$GPRMC,015540.220,9.00,N,0000E,0.0000212,,7
```

Is good? :S

7.

[Mikal](#)

1 week ago

@Gatech,

The sentences TinyGPS is interested in are \$GPGGA and \$GPRMC. In your short example, the \$GPRMC sentence is truncated—they all should end with a *XX checksum—and the \$GPGGA sentence seems slightly bogus. The sixth term is 0, meaning that the data should be considered suspect, and the data itself is a little screwy: Lat = 89 degrees 60 minutes N by 0 degrees 0 minutes E. That's the north pole! I think your GPS hasn't detected a fix yet, but TinyGPS is probably processing this data correctly.

8.

Gatech

1 week ago

Thanks. You know how to make it detect a fix?. Or to know its status?.

Regards.

9.

Gatech

1 week ago

THANKS. I FIXED THIS.

The cables I were using, didn't make a correct connection, and it was in short. :S I checked all with testers, and it was... now is working. Thanks to god it didn't break my GPS.

2. The new Script for v12 is too long and I can't install on my Arduino 😞

10.

Gatech

1 week ago

Now, I can't understand what's happening.

Program tells me some LAT/LONG, and they are really near between them, but looking in Google Maps they are like 1KM far of my real location. :S

Its not supposed to be max some METTERS exactly?.

11.

[Mikal](#)

1 week ago

@Gatech,

Hmmm, my EM-406A is always very close to where Google Maps places it, especially when there are lots of satellites in the equation. The EM-406A I believe advertises +/- 10m accuracy?

12.

Gatech

1 week ago

Thanks. After someeeeeeeeeee... tests, now it's working well!!!.

A question. Is there a method for importing a version of TinyGPS more 'light'? In version v12 test_with_gps_device can't upload on my Arduino due to its size. V11 is lighter, but I don't know if are functionalities on 12 that are important. On my final work I think I only need LAT, LONG and "Precision". Which version of TinyGPS should I use?. My sketch will also have GPRS, and soome code, so I need to use well my 13Kb 😊

Thanks for all your support!.
Gatech. From Colombia.

13.

Mitch

6 days ago

Hi, I'm getting this error when I compile simple_test - "TinyGPS does not name a type", and when I compile test_with_gps_device, a compile error, "variable or field 'gpsdump' declared void" The version 11 examples compiled with no errors. I'm using the latest 1.0 version. Any ideas? Thanks in advance.

14.

David (D9W)

5 days ago

May I make a suggestion?

Background: I am using the new Arduino 1.0 IDE.

I know this is all nuances but I the way you have the code here on your web page>>

```
#define RXPIN 3
#define TXPIN 2 // From the GPS to the Arduino
SoftwareSerial nss(RXPIN, TXPIN);
```

VS the code in the example that comes with TinyGPU (test_with_gps_device)>>

```
/* This sample code demonstrates the normal use of a TinyGPS object.
It requires the use of SoftwareSerial, and assumes that you have a
4800-baud serial GPS device hooked up on pins 2(rx) and 3(tx). (I changed the pin numbers for this example- but not having them name can lead to mistakes).
*/
```

```
TinyGPS gps;
SoftwareSerial nss(2, 3);
```

I think the first example would lead to fewer mistakes of which Pin is the RX and which Pin is the TX, then your second example. I had been fighting with why I could not the TinyGPS to work at all, and it had to do with the fact I had them switch and did not know it. I know this is all code/nuances and the #define statements really don't mean anything but having those named like in your first example made me stop and look how I had things wired up. (Thanks!). Too bad the example can't be updated to reflect the first example. That still does not explain why SoftwareSerial is spitting out so much junk for 1.0 IDE. And I hope you can help me figure out why SoftwareSerial is giving me so many problems (over on the SoftwareSerial web page.)

15.

Ratheesh

4 days ago

Where can I download the library?

16.

Seriogal

1 day ago

Hello,

just tried to move from Arduino v0.23 to v1.0 by changing TinyGPS library (tried v11 and v12). Several remarks:

- print_date(gps) function always return 0/0/2000 and time returns HOUR:MINUTES:SECONDSHUNDREDS (why not seconds only?)
- gps.f_speed_kmph() never return speed
- gps.f_course() never return course

rest functions seems to be working Ok.

P.S. With v10 all these functions worked correctly.

17.

[Mikal](#)

1 day ago

Mitch–

My guess is that there's some kind of an installation problem. Have you figured it out?

18.

[Mikal](#)

1 day ago

@David,

Good suggestion. Thanks!

19.

[Mikal](#)

1 day ago

@Ratheesh,

It's right on this page–though a little hard to see. Sorry!

20.

[Mikal](#)

1 day ago

@Seriogal,

Hmm. These all work for me. Can you try the new “test_with_gps_device” sample? Here's my output:

Sats HDOP Latitude Longitude Fix Date Time Date Alt Course Speed Card Distance Course Card Chars Sentences Checksum
(deg) (deg) Age Age (m) --- from GPS ---- ---- to London ---- RX RX Fail

```
-----
6 240 30.34266 -97.92635 249 02/18/2012 18:31:14 280 233.30 10.73 0.89 N 7919 40.66 NE 502 3 0
7 180 30.34265 -97.92635 142 02/18/2012 18:31:16 162 231.60 38.77 0.85 NE 7919 40.66 NE 793 6 0
7 180 30.34263 -97.92635 354 02/18/2012 18:31:17 373 230.00 195.36 2.41 SSW 7919 40.66 NE 996 8 0
7 180 30.34262 -97.92635 566 02/18/2012 18:31:18 586 228.60 189.42 1.11 S 7919 40.66 NE 1199 10 0
6 220 30.34259 -97.92637 161 02/18/2012 18:31:20 181 226.00 191.97 4.07 SSW 7919 40.66 NE 1790 14 0
7 180 30.34257 -97.92637 464 02/18/2012 18:31:21 484 224.80 188.56 3.22 S 7919 40.66 NE 1993 16 0
7 180 30.34256 -97.92637 582 02/18/2012 18:31:22 602 224.80 183.75 1.43 S 7919 40.66 NE 2220 18 0
6 240 30.34257 -97.92637 204 02/18/2012 18:31:24 232 225.00 171.61 1.52 S 7919 40.66 NE 2632 21 0
6 240 30.34257 -97.92637 486 02/18/2012 18:31:25 506 225.50 169.60 1.07 S 7919 40.66 NE 2990 24 0
6 240 30.34258 -97.92635 704 02/18/2012 18:31:27 92 226.80 84.10 0.44 E 7919 40.66 NE 3314 27 0
7 180 30.34259 -97.92635 159 02/18/2012 18:31:28 180 227.80 189.78 1.17 S 7919 40.66 NE 3594 30 0
7 180 30.34261 -97.92634 126 02/18/2012 18:31:29 146 229.30 80.06 0.56 E 7919 40.66 NE 3987 32 0
7 180 30.34262 -97.92634 148 02/18/2012 18:31:31 178 230.80 186.01 1.06 S 7919 40.66 NE 4383 35 0
-----
```

[« Older Comments](#)

20 Trackbacks For This Post

1. [เริ่มต้นสร้าง GPS จอสีกับอาต๋อโน้ | Ayarafun Factory](#) →
[March 2nd, 2009 → 1:47 pm](#)

[...] http://www.sundial.org/arduino/?page_id=3 [...]

2. [IO](#) →
[March 14th, 2009 → 11:57 am](#)

[...] won't. I answer a few questions on the Arduino microcontroller forum and post an update to a library I [...]

3. [The Hired Gun » GPS project: show me something](#) →
[June 9th, 2009 → 3:08 pm](#)

[...] this journey I also looked into using the TinyGPS library. While it does a nice job of handling the core GPS parsing, I still had the primary desire to log [...]

4. [GPS mit Arduino - Webmeister Blog](#) →
[September 20th, 2009 → 9:05 am](#)

[...] serielle Verbindung werden die Daten ans Display geschickt. Verwendet werden die Bibliotheken TinyGPS und [...]

5. [The Frustrumantic Box, Part 4: Software « New Bright Idea](#) →
[January 20th, 2010 → 1:08 am](#)

[...] Arduino developers for the great librarires, and to Mikal Hart in particular for his work on the TinyGPS and NewSoftSerial [...]

6. [Transmission success - Projects](#) →
[July 25th, 2010 → 6:03 am](#)

[...] and sending the bits of "Hello World" in 7N1 format. I spent some time with the TinyGPS and NewSoftSerial libraries from Mikal Hart, and got the parsing working nicely and building an [...]

7. [GPS testing with LCD Character Display](#) →
[July 25th, 2010 → 11:33 am](#)

[...] can have the LCD output latitude, longitude, or whatever. You'll need the TinyGPS library from Arduiniana downloaded and installed for it to work. They suggest using NewSoftSerial, but I couldn't get [...]

8. [Arduino GPS — not exactly pocket-sized, but cool! « Arduino Projects FTW](#) →
[September 7th, 2010 → 7:40 pm](#)

[...] EEPROM so they are persistent between power cycles. The sketch uses Mikal Hart's excellent TinyGPS library and includes code from the ArduPilot Project for [...]

9. [A fully functional Arduino GPS receiver | Embedded projects from around the web](#) →
[September 11th, 2010 → 2:16 am](#)

[...] uses a TinyGPS library for easier GPS shield access. So device can be used as normal GPS tracking device the only [...]

10. [Moving Forward with Arduino – Chapter 17 – GPS « tronixstuff](#) →
[September 20th, 2010 → 12:12 am](#)

[...] point you will need to install two libraries into the Arduino software – NewSoftSerial and TinyGPS. Extract the folders into the libraries folder within your arduino-001x [...]

11. [Shield Driven Arduino GPS](#) →
[October 7th, 2010 → 10:43 am](#)

[...] sketch uses Mikal Hart's excellent TinyGPS library and includes code from the ArduPilot Project for [...]

12. [GPS bövítés « Nikon D5000 DSLR](#) →
[February 15th, 2011 → 9:06 am](#)

[...] 2 [...]

13. [The Maritime Geocaching Association » Build Your Own Steampunk GPS](#) →
[March 2nd, 2011 → 7:52 am](#)

[...] attached code should pretty much speak for itself. Using the TinyGPS library, your current position is taken and the direction to the final location is calculated. [...]

14. [Moving Forward with Arduino – Chapter 19 – GPS part II « tronixstuff](#) →
[May 15th, 2011 → 10:43 pm](#)

[...] pull-down resistor). You will need to install the SdFAT library, NewSoftSerial library, TinyGPS library and the SdFat library if not already [...]

15. [Arduinoによる放射線データ収集\(3\) « stastaka's Blog](#) →
[June 11th, 2011 → 10:23 pm](#)

[...] <http://arduiniana.org/libraries/tinygps/> USBシリアルを使っているとTX,RXが使えませんが、 [...]

16. [reptile-addict.nl | My Arduino Blog](#) →
[June 21st, 2011 → 12:20 am](#)

[...] Mikal Hart's tinyGPS library; [...]

17. [Communicating to GPS Receiver using USB Host Shield « Circuits@Home](#) →
[July 1st, 2011 → 10:33 pm](#)

[...] how to send raw GPS output to a NMEA 0183 message parser. For the following code example I used Mikal Hart's TinyGps library. Since the library itself is not handling serial input, it was only necessary to make changes in [...]

18. [GpsBee and Seeeduino Stalker v2 « Wireless Building Automation](#) →
[September 20th, 2011 → 11:45 pm](#)

[...] easy developing of the necessary Arduino application I used an existent dedicated library: TinyGPS (<http://arduiniana.org/libraries/tinygps/>), which help us to parse all the packets received on the serial port from the GpsBee module. Ok, so [...]

19. [SheekGeek » Blog Archive » Weather Balloon Payload Testing on a Model Rocket \(Pt.1\)](#) →
[November 20th, 2011 → 4:37 pm](#)

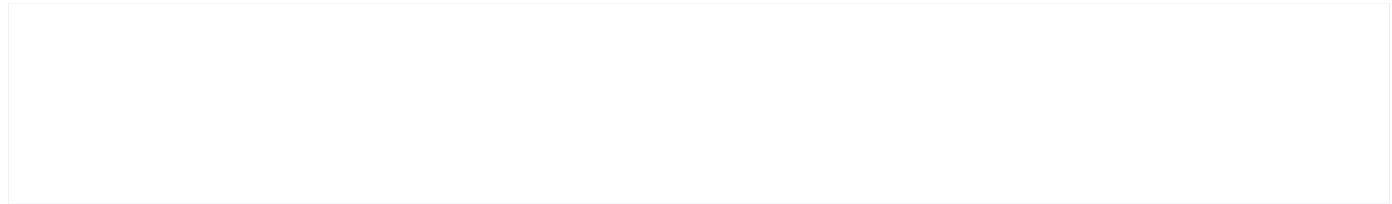
[...] and gyroscope package and slapped together a simple SD card interface. The libraries I used were TinyGPS and fat16lib (for SD card use). Weather_Balloon_Code and schematic in case you'd like to [...]

20. [Moving Forward with Arduino – Chapter 19 – GPS part II « tronixstuff](#) →
[January 30th, 2012 → 4:57 pm](#)

[...] forget the 10k ohm pull-down resistor). You will need to install NewSoftSerial library, TinyGPS library and the SdFat library if not already [...]

Leave a Reply

	Name *
	Mail *
	Website

[Add Comment](#)

Pages

- [About](#)
- [Libraries](#)
 - [Flash](#)
 - [NewSoftSerial](#)
 - [PString](#)
 - [PWMServo](#)
 - [Streaming](#)
 - [TinyGPS](#)
- [Projects](#)
 - [The Reverse Geocache™ Puzzle](#)
 - [Building a Puzzle Box](#)
 - [More Puzzle Box Tales](#)

Recent Posts

- [Puzzle Box: The Poetry of the Quest](#)
- [Engagements: A Tale of Two Cities](#)
- [Welcome, MAKE readers!](#)
- [NewSoftSerial 11 \(beta\)](#)
- [Puzzle Box 4: A Long Awaited Opening](#)

Archives

- [January 2012](#)
- [January 2011](#)
- [October 2010](#)
- [August 2010](#)
- [July 2010](#)
- [February 2010](#)
- [January 2010](#)
- [November 2009](#)
- [October 2009](#)
- [May 2009](#)
- [April 2009](#)
- [March 2009](#)
- [February 2009](#)
- [January 2009](#)

© 2012 Mikal Hart.