

GPS Receiver for Arduino (Model A) (SKU:TEL0083-A)

From Robot Wiki

Contents

- 1 Introduction
- 2 Specifications
- 3 Line Definition
- 4 Tutorial
- 5 GPS with LCD
Sample Code
- 6 Documents



Introduction

GPS Receiver for Arduino is an unit embedding GPS module and antenna in a small foot-print enclosure. By using TinyGPS library, Arduino can retrieve geographic coordinates (latitude and longitude, altitude), speed, heading and GMT time.

The update rate is an important performance index of a GPS receiver. Most GPS in mobile phones provide an update rate of 1Hz, which means, only one set of data can be retrieved in one second. For GPS receivers with 5Hz, the data interval is much reduced and thus can be used for more demanding applications (e.g. on fast-moving vehicles)

Specifications

- U-BLOX G6010 solution
- 5Hz output
- 38400bps TTL serial interface
- 3.3V-5V input voltage

- 50-channel receiver
- Extremely high sensitivity -161dBm
- Accuracy 2.5m (Autonomous) / <2m[SBAS]
- Operating temperature: -40°C to 85°C
- Power consumption: 3.3V @ 41mA
- Hot Start : 1s
- Warm Start : 32s
- Cold Start : 32s
- Ceramic Antena 25*25*2mm
- Module Size 25*28*7.5mm
- Enclosure Size 40x30x10mm

Line Definition

- Black: RxD
- White: TxD
- Red: VCC
- Black (thick): GND

Tutorial

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:

```
void loop()
{
  while (Serial.available())
  {
    int c = Serial.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();
```

GPS with LCD Sample Code



GPS receiver Arduino

VCC ————— VCC

GND ————— GND

RX ————— TX

TX ————— RX

Library: TinyGPS V1.3 (<https://github.com/mikalhart/TinyGPS/archive/v1.3.zip>)

```

?
1 #include <TinyGPS.h>
2 #include <LiquidCrystal.h>
3
4 TinyGPS gps;
5 LiquidCrystal lcd(8, 9, 4, 5, 6, 7);           //LCD driver pins
6 int led = 13;
7
8 long lat, lon;
9 unsigned long fix_age, time, date, speed, course;
10 unsigned long chars;
11 unsigned short sentences, failed_checksum;
12 //int year;
13 //byte month, day, hour, minute, second, hundredths;
14
15 int DEG;
16 int MIN1;
17 int MIN2;
18
19 void LAT() {                                //Latitude state
20     DEG=lat/1000000;
21     MIN1=(lat/10000)%100;
22     MIN2=lat%10000;
23
24     lcd.setCursor(0,0);                    // set the LCD cursor position
25     lcd.print("LAT:");
26     lcd.print(DEG);
27     lcd.write(0xDF);
28     lcd.print(MIN1);
29     lcd.print(".");
30     lcd.print(MIN2);
31     lcd.print("'  ");
32 }
33 void LON() {                                //Longitude state
34     DEG=lon/1000000;
35     MIN1=(lon/10000)%100;
36     MIN2=lon%10000;
37
38     lcd.setCursor(0,1);                    // set the LCD cursor position
39     lcd.print("LON:");
40     lcd.print(DEG);
41     lcd.write(0xDF);
42     lcd.print(MIN1);
43     lcd.print(".");
44     lcd.print(MIN2);

```

```
45  lcd.print("  ");
46}
47
48
49
50void setup()
51{
52  Serial.begin(38400);          //Set the GPS baud rate.
53
54  pinMode(led, OUTPUT);
55
56  lcd.begin(16, 2);            // start the library
57  lcd.setCursor(0,0);          // set the LCD cursor position
58  lcd.print("GPS test");       // print a simple message on the LCD
59  delay(2000);
60}
61
62void loop()
63{
64  while (Serial.available())
65  {
66    digitalWrite(led, HIGH);
67    int c = Serial.read();      // Read the GPS data
68    if (gps.encode(c))         // Check the GPS data
69    {
70      // process new gps info here
71    }
72
73  }
74  digitalWrite(led, LOW);
75  gps.get_position(&lat, &lon, &fix_age); // retrieves +/- lat/long in 100000ths
76of a degree
77  gps.get_datetime(&date, &time, &fix_age); // time in hhmmsscc, date in ddmmyy
78
79  //gps.crack_datetime(&year, &month, &day, //Date/time cracking
80  //&hour, &minute, &second, &hundredths, &fix_age);
81
82LAT();
83LON();
84
  }
```

Documents

TinyGPS (<http://arduiniana.org/libraries/tinygps/>)

U-Center Download (<http://www.u-blox.com/en/evaluation-tools-a-software/u-center/u-center.html>)

U-Center User Guide ([http://www.u-blox.com/images/downloads/Product_Docs/u-Center_User_Guide_\(UBX-13005250\).pdf](http://www.u-blox.com/images/downloads/Product_Docs/u-Center_User_Guide_(UBX-13005250).pdf))

Retrieved from "[http://www.dfrobot.com/wiki/index.php?](http://www.dfrobot.com/wiki/index.php?title=GPS_Receiver_for_Arduino_(Model_A)_(SKU:TEL0083-A)&oldid=26731)

[title=GPS_Receiver_for_Arduino_\(Model_A\)_\(SKU:TEL0083-A\)&oldid=26731](http://www.dfrobot.com/wiki/index.php?title=GPS_Receiver_for_Arduino_(Model_A)_(SKU:TEL0083-A)&oldid=26731)"

Categories: [Product Manual](#) | [TEL Series](#) | [Wireless](#)

- This page was last modified on 18 June 2014, at 14:23.
- This page has been accessed 921 times.