Parsing NMEA Sentences

There are many sentences in the NMEA standard, the most common ones are:

- $GPRMC (Global Positioning Recommended Minimum Coordinates) provides the time, date, latitude, longitude, altitude and estimated velocity.
- $GPGGA sentence provides essential fix data which provide 3D location and accuracy data.

Let's take an example of $GPRMC NMEA sentence from a GPS receiver.

$GPRMC, 123519, A, 4807.038, N, 01131.000, E,022.4, 084.4, 230394, 003.1, W*6A

$	Every NMEA sentence starts with $ character.
GPRMC	Global Positioning Recommended Minimum Coordinates
123519	Current time in UTC – 12:35:19
A	Status A=active or V=Void.
4807.038,N	Latitude 48 deg 07.038′ N
01131.000,E	Longitude 11 deg 31.000′ E
022.4	Speed over the ground in knots
084.4	Track angle in degrees True
220318	Current Date – 22rd of March 2018
003.1,W	Magnetic Variation
*6A	The checksum data, always begins with *

Let's take an example of $GPGGA NMEA sentence.

$GPGGA, 123519, 4807.038, N, 01131.000, E, 1, 08, 0.9, 545.4, M, 46.9, M, , *47

$	Starting of NMEA sentence.

GPGGA	Global Positioning System Fix Data
123519	Current time in UTC – 12:35:19
4807.038,N	Latitude 48 deg 07.038′ N
01131.000,E	Longitude 11 deg 31.000′ E
1	GPS fix
08	Number of satellites being tracked
0.9	Horizontal dilution of position
545.4,M	Altitude in Meters (above mean sea level)
46.9,M	Height of geoid (mean sea level)
(empty field)	Time in seconds since last DGPS update
(empty field)	DGPS station ID number
*47	The checksum data, always begins with *

For more information about NMEA sentences and what data they contain, check out gpsinformation.org

Arduino Code – TinyGPS Library

Often for our projects, we need to parse NMEA sentences into useful information. To simplify our work, we have a library called TinyGPS++ library.

This library does a lot of heavy lifting required for receiving data from GPS modules, such as reading and extracting useful data in the background. So, we don't need to worry about icky parsing work.

Thanks to Mikal Hart for his great contribution. His website Arduiniana.org has a full overview of all of the capabilities of the TinyGPS++ library.

Download the library first, by visiting the GitHub repo or, just click this button to download the zip:

TinyGPSPlus.zip

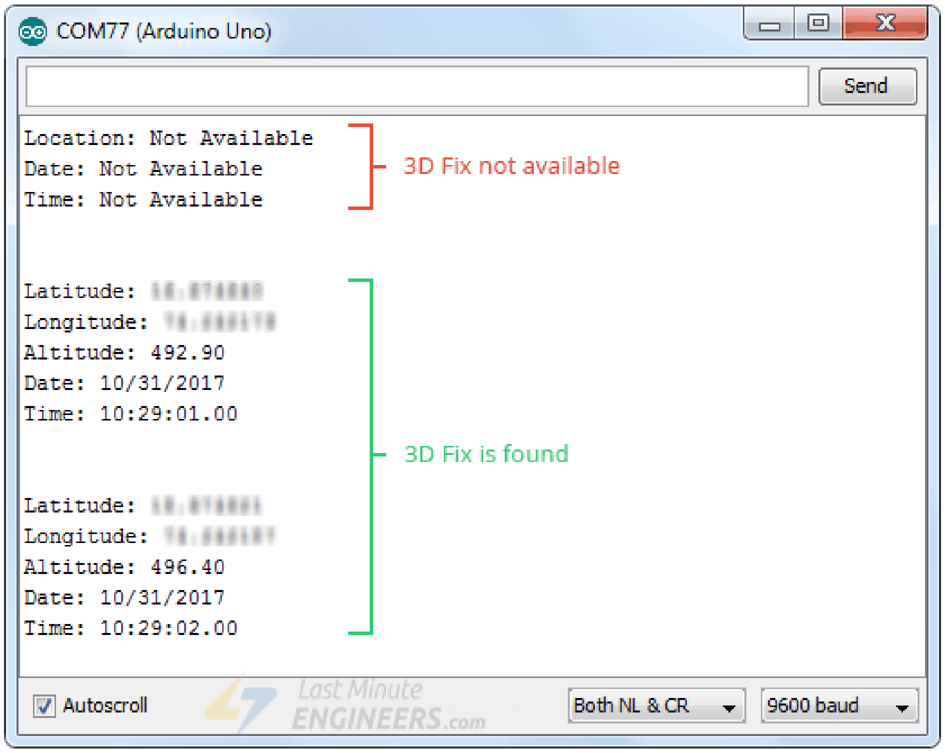
To install it, open the Arduino IDE, go to Sketch > Include Library > Add .ZIP Library, and then select the TinyGPSPlus ZIP file that you just downloaded. If you need more details on installing a library, visit this Installing an Arduino Library tutorial.

Once you have the library installed, you can copy below sketch into the Arduino IDE.

The following test sketch will print the location information(Latitude, Longitude & Altitude) and UTC(Date & Time) on the serial monitor. Try the sketch out; and then we will explain it in some detail.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

// Choose two Arduino pins to use for software serial
int RXPin = 2;
int TXPin = 3;

int GPSBaud = 9600;

// Create a TinyGPS++ object
TinyGPSPlus gps;

// Create a software serial port called "gpsSerial"
SoftwareSerial gpsSerial(RXPin, TXPin);

void setup()
{
  // Start the Arduino hardware serial port at 9600 baud
  Serial.begin(9600);

  // Start the software serial port at the GPS's default baud
  gpsSerial.begin(GPSBaud);
}

void loop()
{
  // This sketch displays information every time a new sentence is correctly encoded.
```

This is how the output looks like on the serial monitor.



```
Location: Not Available
Date: Not Available
Time: Not Available


Latitude: [redacted]
Longitude: [redacted]
Altitude: 492.90
Date: 10/31/2017
Time: 10:29:01.00


Latitude: [redacted]
Longitude: [redacted]
Altitude: 496.40
Date: 10/31/2017
Time: 10:29:02.00
```

Code Explanation:

The sketch starts by including TinyGPS++ library and software serial library. Then, we define arduino pins to which NEO-6M GPS module is connected and variable that stores default GPS baud rate.

Creating `TinyGPSPlus` object will help access special functions related to the library. Next, we create a software serial port called `gpsSerial` through which we can talk to the module.

```
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

int RXPin = 2;
int TXPin = 3;
int GPSBaud = 9600;
```

```
TinyGPSPlus gps;
SoftwareSerial gpsSerial(RXPin, TXPin);
```

In the `setup` function, we need to initiate the serial communication with the PC as well as the GPS module.

```
void setup()
{
  Serial.begin(9600);
  gpsSerial.begin(GPSBaud);
}
```

In `loop` function, we call `displayInfo()` custom function which prints location information(Latitude, Longitude & Altitude) and UTC(Date & Time) on the serial monitor, every time a new NMEA sentence is correctly encoded.

If 5000 milliseconds pass and there are no characters coming in over the software serial port, we show No GPS detected error.

```
void loop()
{
  while (gpsSerial.available() > 0)
    if (gps.encode(gpsSerial.read()))
      displayInfo();

  if (millis() > 5000 && gps.charsProcessed() < 10)
  {
    Serial.println(F("No GPS detected"));
    while(true);
  }
}
```

Other Useful Functions In TinyGPS++ Library

There are a few useful functions you can use with TinyGPS++ object. Few of them are listed below:

- `gps.speed.value()` function returns current ground speed in 100ths of a knot.
- `gps.course.value()` function returns current ground course in 100ths of a degree.
- `gps.satellites.value()` function returns the number of visible, participating satellites.
- `gps.hdop.value()` function returns horizontal diminution of precision.
- If you want to know how old an object's data is, call its `age()` method, which returns the number of milliseconds since its last update. If this returns a value greater than 1500 or so, it may be a sign of a problem like a lost fix.
- If you want to extract data from any other NMEA sentence. You can use library's custom extraction functionality by telling TinyGPS++ the sentence name and the field number you are interested in, like this: `TinyGPSCustom magneticVariation(gps, "GPRMC", 10)` And you can query it just like the others: `magneticVariation.value()`

U-center software

U-center from u-blox is a powerful tool for evaluation, performance analysis and configuration of u-blox GPS receivers including NEO-6M. It's a free tool but can only be used on Windows platform.

It can display realtime structured and graphical data visualization from any GPS receiver such as

- Satellite summary view
- Navigation summary view
- Compass, speedometer, clock, altimeter
- Chart view of any two parameters of choice
- Data recording and playback functionality

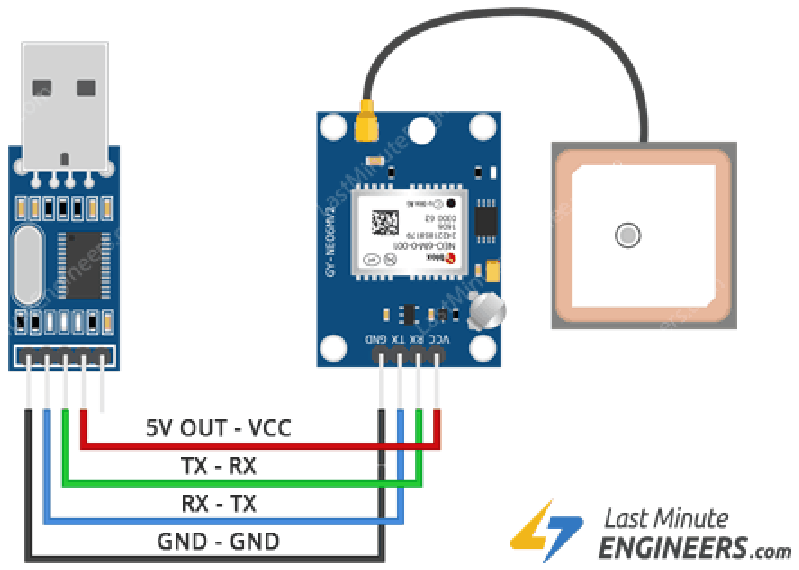
The software can be downloaded from U-blox website.

U-Center

Connecting NEO-6M to U-center

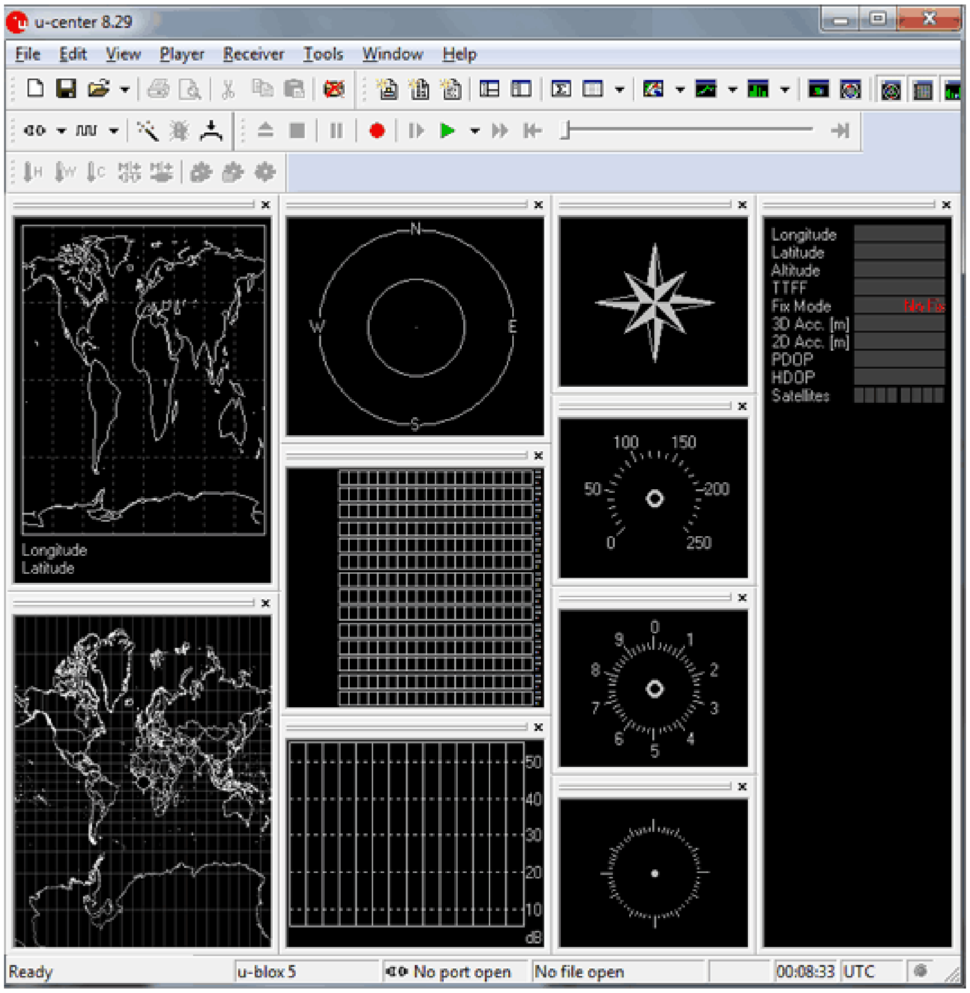
In order to use U-center software, you need to connect your NEO-6M to PC using any USB to TTL converter.

Below image shows NEO-6M connected to PC through PL2303 USB to TTL converter.

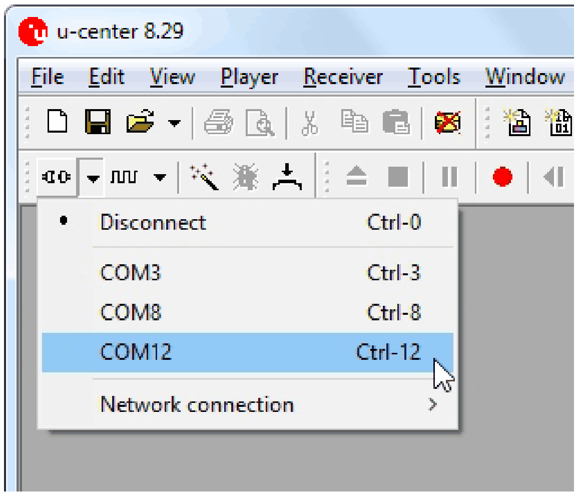


Using U-center

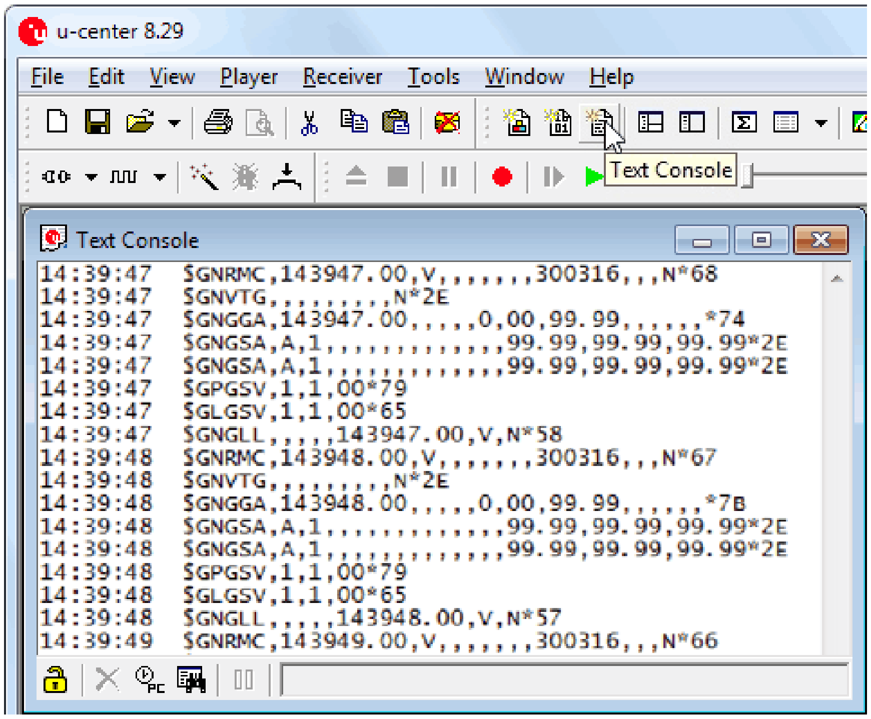
After a successful installation, u-center can be started from the Start Menu (All Programs -> u-blox-> u-center -> u-center) and will start up as shown below.



Locate the communication toolbar and click on the arrow beside the icon. This will show a list with all available COM ports. Select the corresponding COM port where the receiver is connected.

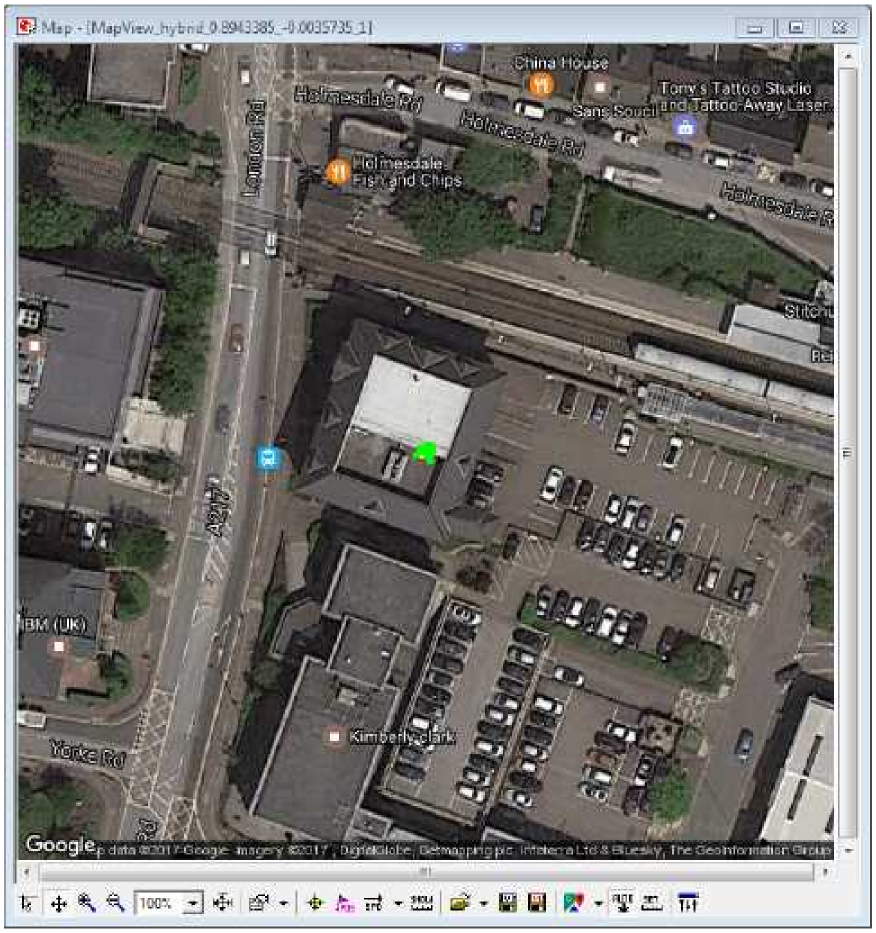


The text console button will show you the raw NMEA sentences. This is handy for quickly inspecting the visible ASCII coming from the module over USB.



```
14:39:47  $GNRMC,143947.00,V,,,,,,,300316,,,N*68
14:39:47  $GNVTG,,,,,,,,,N*2E
14:39:47  $GNGGA,143947.00,,,,,0,00,99.99,,,,,,*74
14:39:47  $GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*2E
14:39:47  $GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*2E
14:39:47  $GPGSV,1,1,00*79
14:39:47  $GLGSV,1,1,00*65
14:39:47  $GNGLL,,,,,143947.00,V,N*58
14:39:48  $GNRMC,143948.00,V,,,,,,,300316,,,N*67
14:39:48  $GNVTG,,,,,,,,,N*2E
14:39:48  $GNGGA,143948.00,,,,,0,00,99.99,,,,,,*7B
14:39:48  $GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*2E
14:39:48  $GNGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*2E
14:39:48  $GPGSV,1,1,00*79
14:39:48  $GLGSV,1,1,00*65
14:39:48  $GNGLL,,,,,143948.00,V,N*57
14:39:49  $GNRMC,143949.00,V,,,,,,,300316,,,N*66
```

u-center can display positions on pre-calibrated or Google online (dynamic) maps


Map - [MapView_hybrid_9.8943385_-0.0035735_1]
China House
Holmesdale Rd
Tony's Tattoo Studio and Tattoo-Away Laser
Sans Souci
Holmesdale Fish and Chips
London Rd
A217
IBM (UK)
Kimberly-clark
Yorke Rd
Google
100%

For more information about U-center software, please refer this user guide.