

## Interfacing GY-GPSV3-NEO-M8N GPS Module with Arduino



### Introduction

The NEO-M8N is a GPS module that can be plugged into your Arduino board. It uses satellites to find your location and then sends that information to your Arduino. This lets you add location features to your projects, like a GPS tracker or a robot that navigates based on its position.

This model is newer than the NEO-6M provides more stability and still keeps that same compact form factor. This model also comes with 2 LED indicator, Red LED is for indicating whether the module is turning ON or OFF and Green LED for indicating whether the module already connected to GPS.

### Key Advantages of the Neo-M8N GPS Module Over the Neo-6M:

- **Improved Accuracy:** The Neo-M8N offers a higher accuracy of 2 meters compared to the Neo-6M's 2.5 meters.
- **Faster Update Rate:** The Neo-M8N boasts a faster update rate of 10 Hz, allowing for more frequent position updates than the Neo-6M's 1 Hz.
- **Lower Power Consumption:** The Neo-M8N is designed to be more power-efficient, making it suitable for battery-powered applications.
- **Multiple Constellation Support:** The Neo-M8N can track GPS, GLONASS, Galileo, and BeiDou satellites, providing better coverage and performance in various locations.
- **Newer Technology:** As a more recent product, the Neo-M8N may benefit from advancements in GPS technology and offer additional features or capabilities.

## Specifications

<b>Model</b>	Ublox NEO-M8N
<b>Receiver Type</b>	72-channel Ublox M8 engine
<b>Input Supply Voltage (VDC)</b>	3~5 V
<b>Main Chip</b>	Ublox NEO-M8N
<b>Sensitivity (dBm)</b>	156 dBm
<b>Position Accuracy (Meter)</b>	2 - 2.5m
<b>Acceleration (g)</b>	≤ 4 g
<b>Navigation Update Rate</b>	5 - 10 Hz
<b>Operating Temperature Range (°C)</b>	-45 to 105 °C
<b>Tracking Sensitivity (dBm)</b>	165 – 167 dBm
<b>Co-Ordinate System</b>	WGS-84
<b>Capture Time</b>	0.1s Average
<b>Average Cold Start Time (s)</b>	≤ 29 s
<b>Warm / Hot Start Time (s)</b>	1 s
<b>Boot Time (s)</b>	15 s
<b>Maximum Altitude (m)</b>	50,000 m
<b>Dimensions (mm)</b>	
<b>Antenna</b>	26 x 26 x 9* mm
<b>Module</b>	26 x 36 x 4 mm
<b>Weight (gm):</b>	17g

## Objective

Interfacing the GPS Module with the Arduino (*or with any compatible microcontroller*) and getting the GPS data.

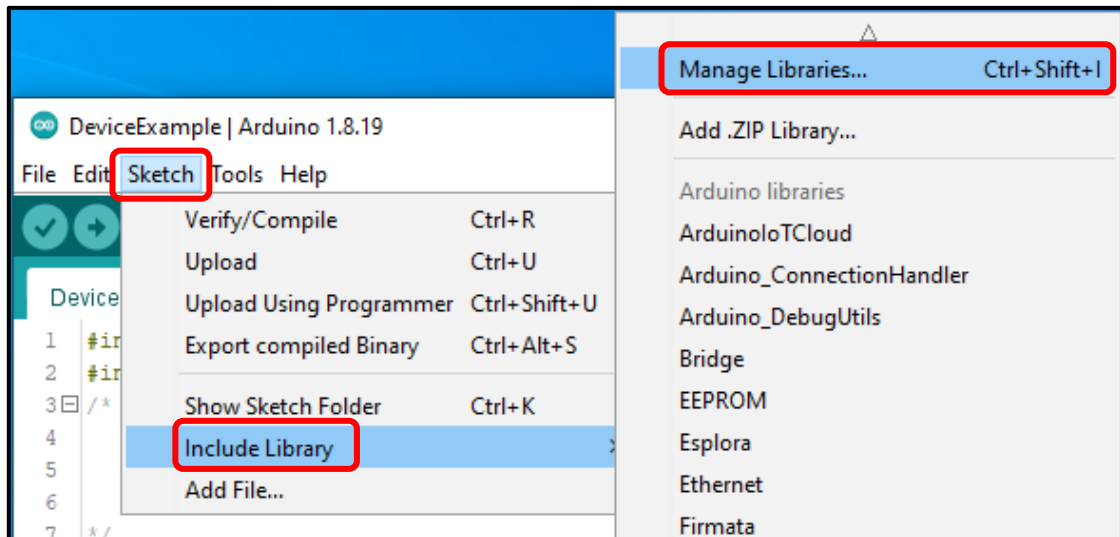
## List of items needed

1. NEO-M8N GPS Module
2. Arduino UNO (any compatible MCUs)
3. Jumper Wires
4. Breadboard

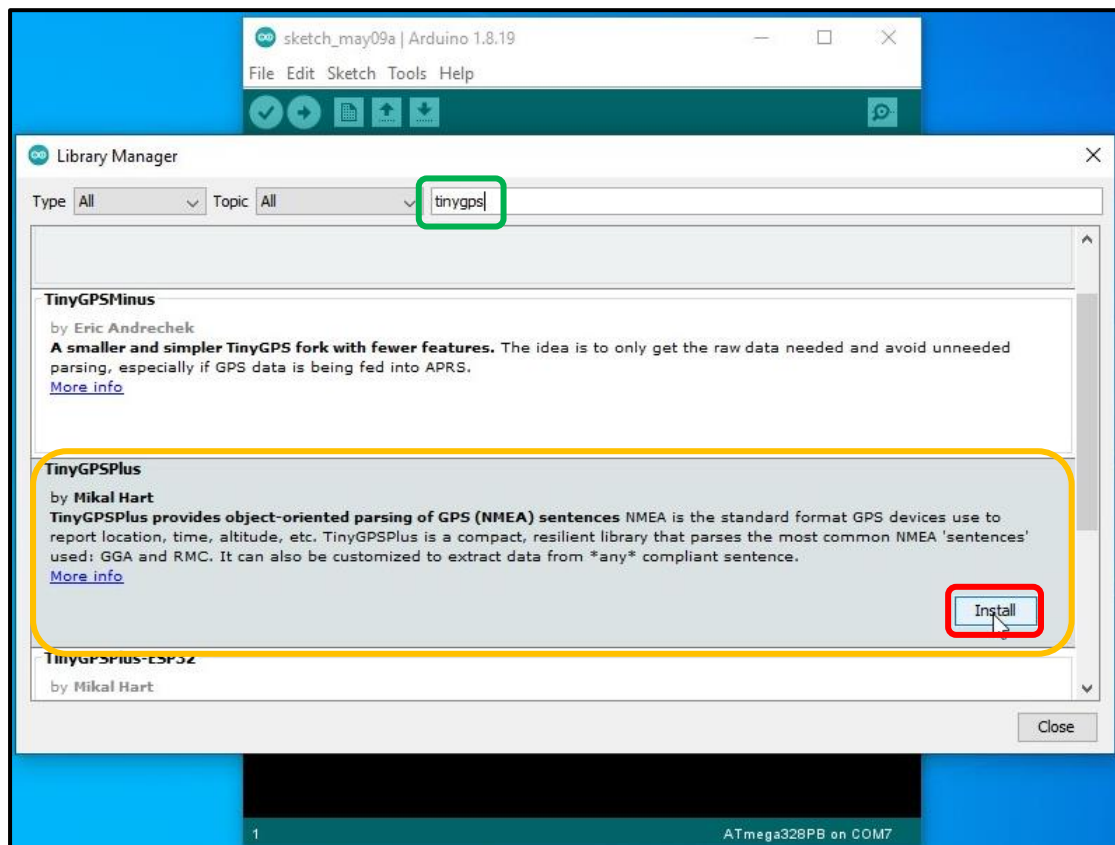
## Procedures

### A. Installing the Arduino library.

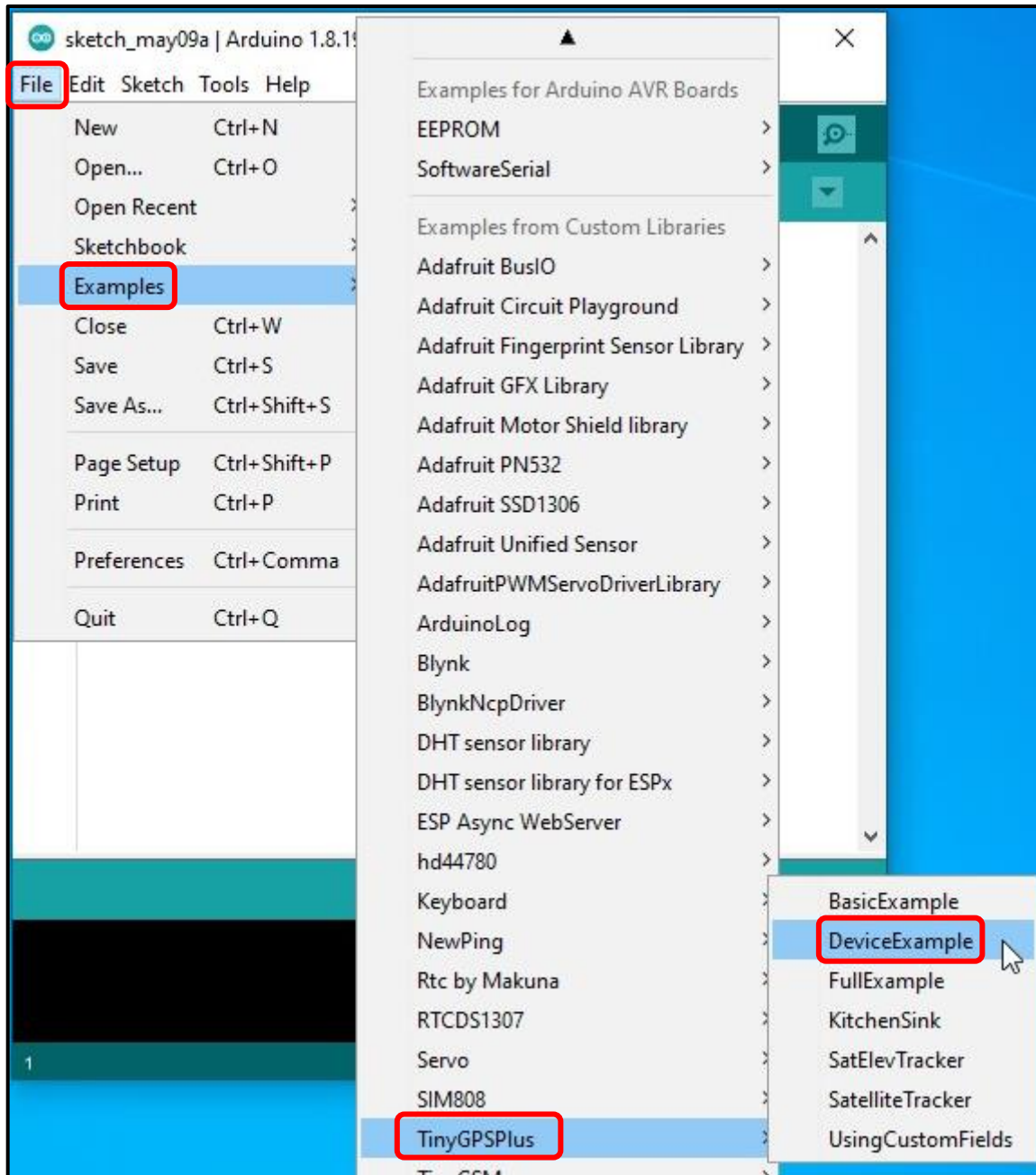
1. Launch **Arduino IDE**, navigate to the **toolbar** and select the **Sketch > Include Library > Manage Libraries....**



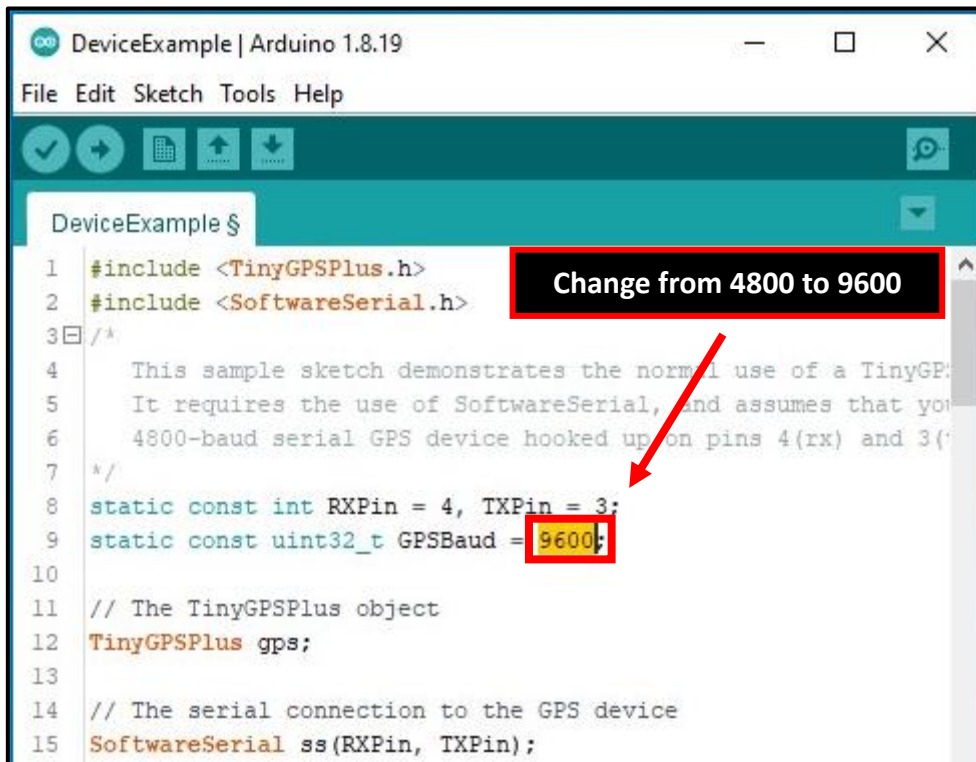
2. Once **Library Manager** window appeared, search for '**TinyGPSPlus**' and click on **Install**.



- Once library downloaded & installed open the code sample code provided. From toolbar, **Examples > TinyGPSPlus > DeviceExample**.



- The code will utilize the **Digital Pin 3 and 4** to receive and send data between both Arduino and the GPS Module.
- Change** the GPS baud rate from **4800** to **9600**. By default, the **NEO-M8N** GPS Module run at **9600** bps from factory.

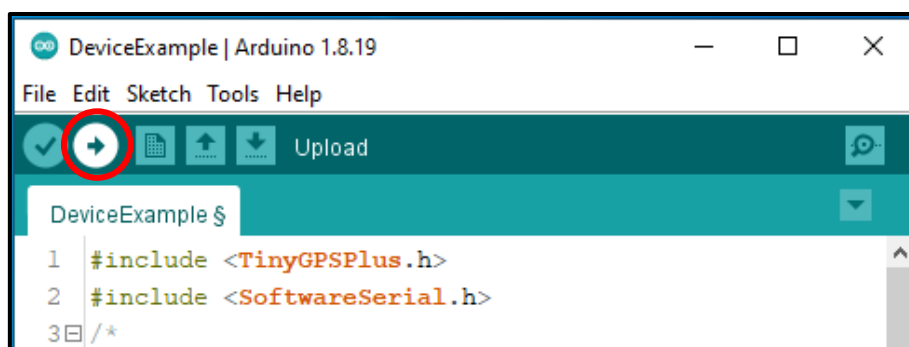


The screenshot shows the Arduino IDE interface with a sketch named 'DeviceExample'. The code is as follows:

```
1 #include <TinyGPSPlus.h>
2 #include <SoftwareSerial.h>
3 /*
4  This sample sketch demonstrates the normal use of a TinyGPS
5  It requires the use of SoftwareSerial, and assumes that you
6  4800-baud serial GPS device hooked up on pins 4(rx) and 3(t
7  */
8 static const int RXPin = 4, TXPin = 3;
9 static const uint32_t GPSBaud = 9600;
10
11 // The TinyGPSPlus object
12 TinyGPSPlus gps;
13
14 // The serial connection to the GPS device
15 SoftwareSerial ss(RXPin, TXPin);
```

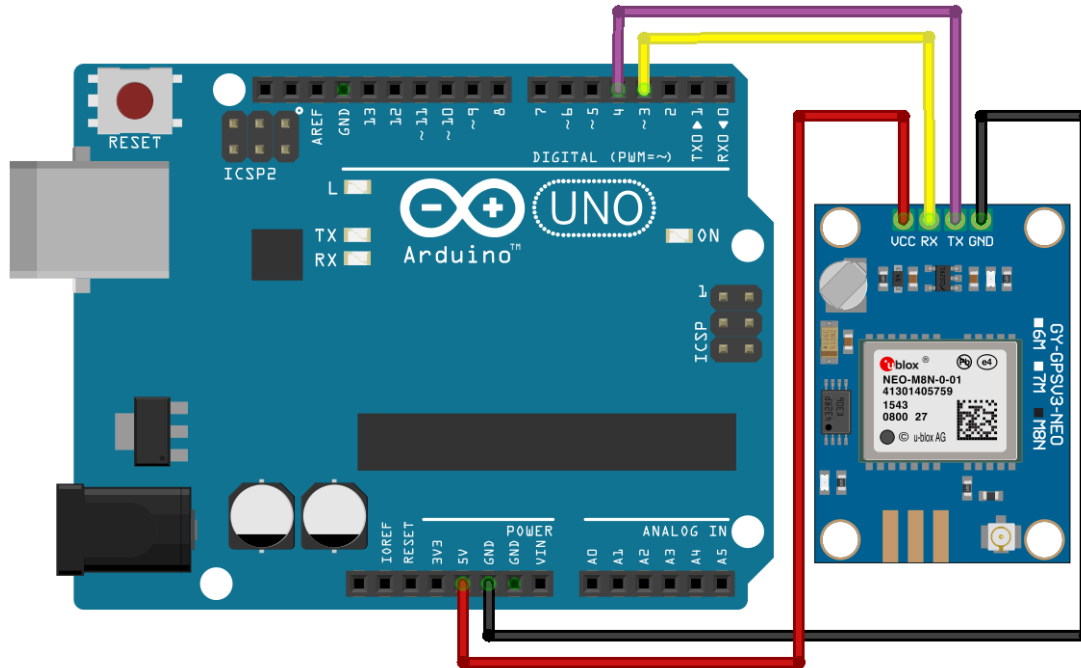
A red box highlights the line `static const uint32_t GPSBaud = 9600;` with the text "Change from 4800 to 9600" and a red arrow pointing to the value 9600.

- Once code correctly configured **Upload** the code to **Arduino UNO** board.



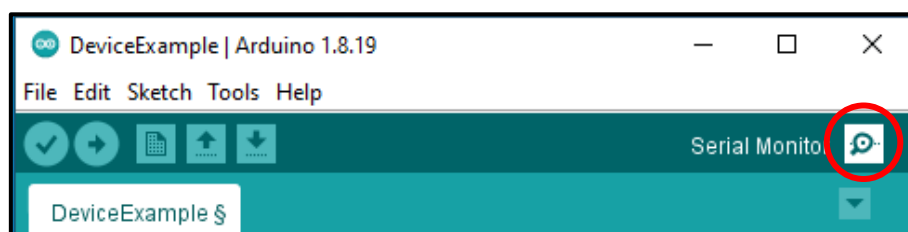
## B. Wiring Setup.

Arduino UNO R3		GY-GPSV3-NEO-M8N	
5V		VCC	
GND		GND	
D4 (as Receiver)	←	TX	
D3 (as Transmitter)	→	RX	



## C. Check the Result.

1. After code uploaded & both boards (*Arduino UNO and GPS Module*) connected, open the **Serial Monitor** to check the acquired GPS data.

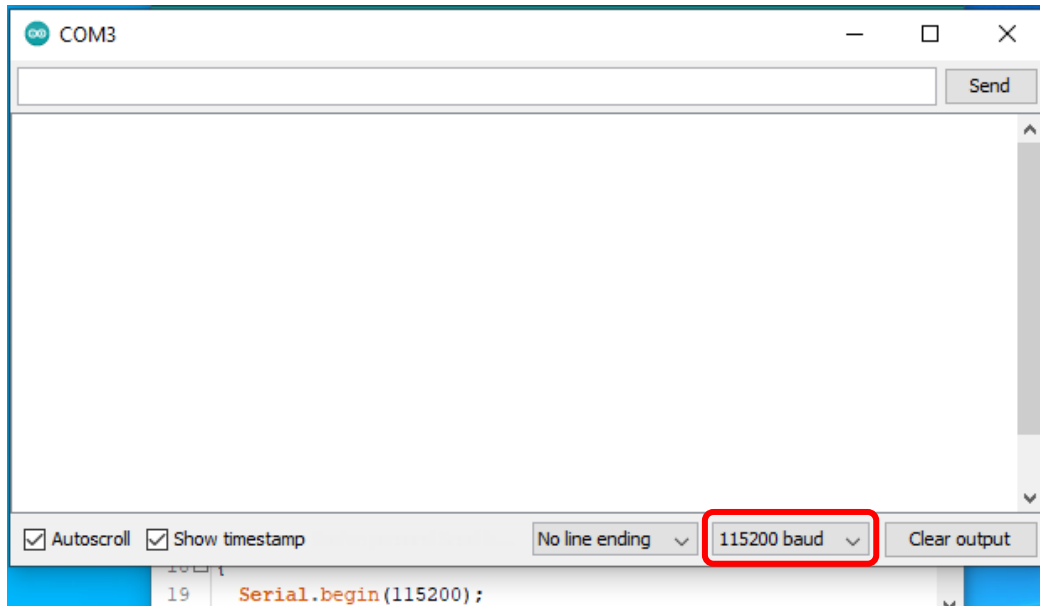




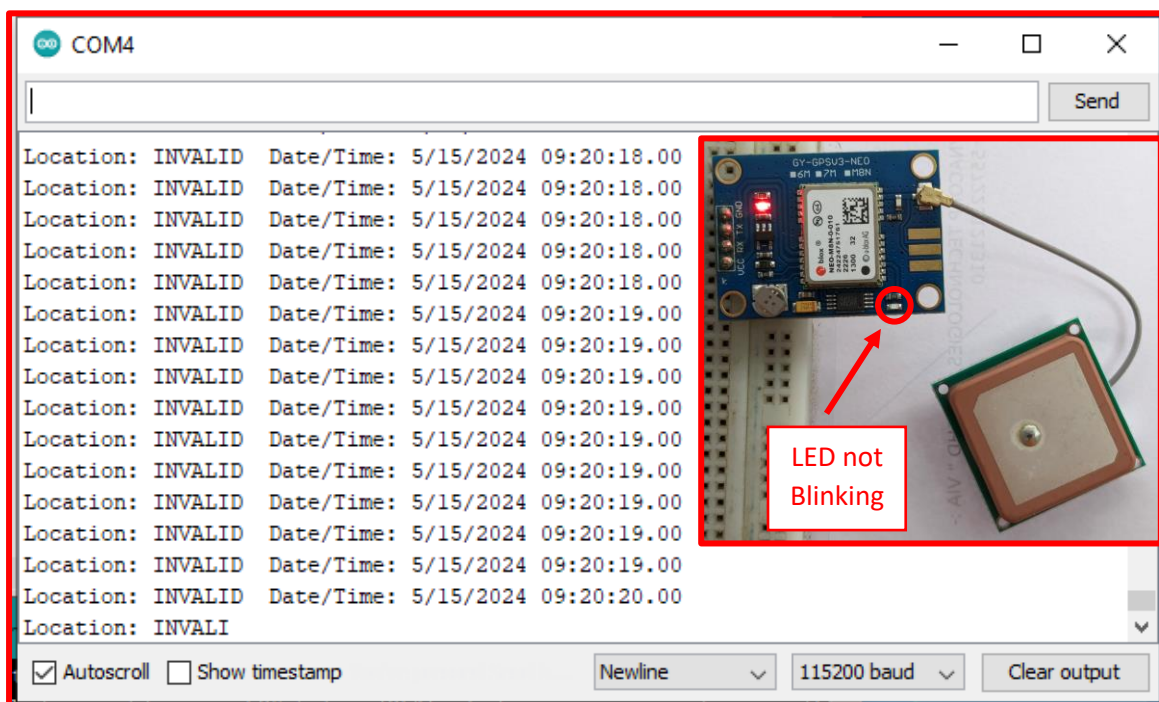
2. Make sure to set the **Serial Baud Rate** same as in the code which is **115200 baud**.  
(Do not get confused with GPS Baud Rate this is for displaying data from Arduino)

(Arduino UNO -> **GPS NEO-M8N = 9600 bps**)

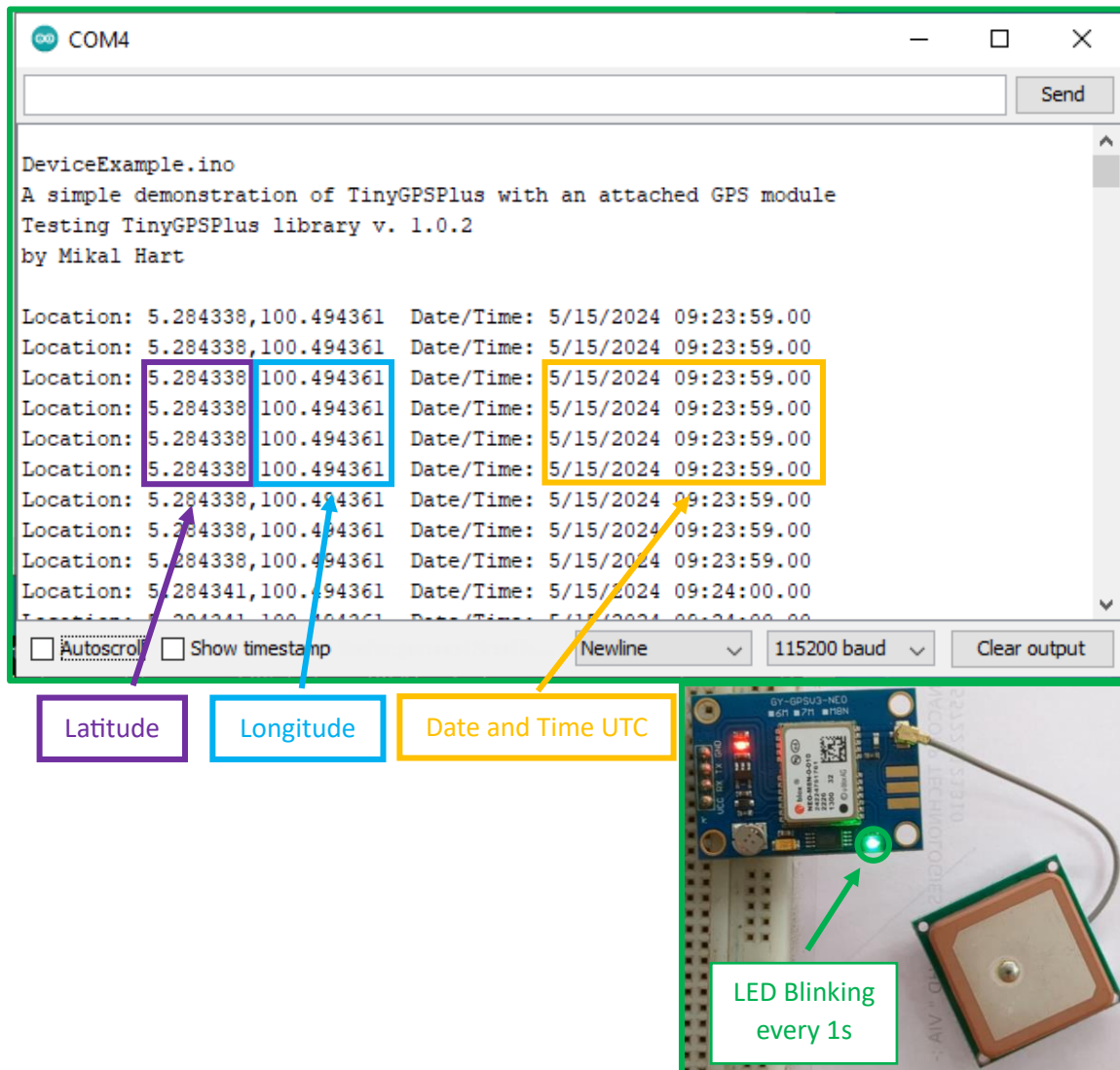
(Arduino UNO -> **Serial Monitor = 115200 bps**)



### GPS Location does not fix



## GPS Location Fixed



COM4

```
DeviceExample.ino
A simple demonstration of TinyGPSPlus with an attached GPS module
Testing TinyGPSPlus library v. 1.0.2
by Mikal Hart

Location: 5.284338,100.494361 Date/Time: 5/15/2024 09:23:59.00
Location: 5.284338,100.494361 Date/Time: 5/15/2024 09:23:59.00
Location: 5.284338,100.494361 Date/Time: 5/15/2024 09:23:59.00
Location: 5.284338,100.494361 Date/Time: 5/15/2024 09:23:59.00
Location: 5.284338,100.494361 Date/Time: 5/15/2024 09:23:59.00
Location: 5.284338,100.494361 Date/Time: 5/15/2024 09:23:59.00
Location: 5.284338,100.494361 Date/Time: 5/15/2024 09:23:59.00
Location: 5.284338,100.494361 Date/Time: 5/15/2024 09:23:59.00
Location: 5.284338,100.494361 Date/Time: 5/15/2024 09:23:59.00
Location: 5.284341,100.494361 Date/Time: 5/15/2024 09:24:00.00
```

Autoscroll Show timestamp Newline 115200 baud Clear output

Latitude Longitude Date and Time UTC

LED Blinking every 1s

## Troubleshooting (What to do if module is not working properly?)

1. **Check the wiring**, make sure the **Receiver, Rx and Transmitter, Tx** connection is correct.
2. Make sure the **Baud rate is correct. (9600 bps for NEO-M8N and 6M)**
3. **Check the Antenna U.FL connector**, make sure the connection is **solid**.
4. **Test** the GPS on the **open space area (outdoor)**. GPS Signal might be **weaker** when tested **Indoor**. (Alternatively, user can upgrade their antenna)
5. **Power** the Arduino or GPS Module with a Sufficient Power Adapter / Supply or Battery (**5v**).
6. **Let the module powered by Arduino for a few minutes** if module isn't used for a long period of times. This module equipped with a small battery to act as capacitor to store configuration data. (**normally that battery discharged let it charge first**)
7. **Test and Debug** the Module **with** manufacturer provided program **U-Center**.