



GNSS WITH DEAD-RECKONING (DR)

GNSS + DR Board User Manual



Purpose of the Document

The purpose of this document is to explain the GNSS board with Dead-Reckoning feature. This document contains the features of GNSS board and how to use it to use it for high accuracy and high-precision positioning.

Document History

Version	Author	Date	Description
A	5G HUB	06.27.2021	Initial Document

Table of Contents

Purpose of the Document	2
Document History	2
1 Package Contents	5
1.1 GNSS Sensor board	5
1.2 Download	5
2 Introduction	6
3 Typical Applications	6
4 GNSS Constellations	7
4.1 GPS	7
4.2 BeiDou	7
4.3 GLONASS	7
4.4 Galileo	7
4.5 IRNSS	7
4.6 QZSS	7
5 Augmentation System	8
5.1 SBAS	8
5.2 AGNSS	8
5.3 Real-Time Kinematic (RTK)	8
5.4 Odometer (ODO)	8
5.5 Geofencing	8
5.6 Dead Reckoning Function	9
6 Hardware Board and Case Diagram	9
7 Connecting using the UART or SPI	10
8 Using Serial Terminal	12
9 DR Configuration	14
9.1 Setting the Orientation	14
10 Mounting	17
11 Dead-Reckoning Calibration	18
12 Enable Dead-Reckoning and 6-axis Sensor	18
13 Sensor Messages	20
14 Using QGNSS Tool	20
15 Using Arduino IDE	21

1 Package Contents

1.1 GNSS Sensor board

- GNSS sensor board

1.2 Download

Arduino sketches for the GNSS board can be downloaded from the following website:

<https://github.com/5ghub/5G-NB-IoT/tree/master/KitSketches>

QNSS Tool is here:

[5G-NB-IoT/Tools at master · 5ghub/5G-NB-IoT \(github.com\)](https://github.com/5ghub/5G-NB-IoT/tree/master/Tools)

To use the board with Arduino IDE and starts running Arduino projects and sketches, install the following software:

Install Arduino IDE for Windows from the following website:

<https://www.arduino.cc/en/Main/Software>

Download and install Arduino library (**5G-NB-IoT_Arduino.zip**) here:

<https://github.com/5ghub/5G-NB-IoT>

2 Introduction

The GNSS board is a compact board for GNSS applications. It features a concurrent multi-constellation GNSS receiver on dual GNSS bands, an integrated 6-axis sensor, fusion with Real-Time Kinematic (RTK) and Dead-Reckoning (DR). The module can achieve sub-meter-level accuracy in open-sky areas.

The GNSS board can work on L1 and L5 bands for GPS, Galileo and QZSS, L1 band for GLONASS and BeiDou, and L5 band for IRNSS. This greatly increases the number of satellites which can be involved in tracking and positioning, thereby significantly reducing the multipath effect from tall buildings in urban environments, reducing signal acquisition time and improving positioning accuracy. In addition to this the GNSS module's on-board LNAs and SAW filters serve to ensure better positioning in weak signal areas and harsh environments.

The GNSS module's combination of compact design, low power consumption and high performance makes it a popular choice for real-time tracking systems for vehicles, people and assets, as well as for sharing economy applications

Feature Highlights

- Support dual-band and multi-constellation
- Supports GPS/BeiDou/GLONASS/Galileo/IRNSS/QZSS
- High-performance, high reliability positioning engine. It facilitates a fast and precise GNSS positioning capability.
- Supports serial communication interfaces UART and SPI.
- Integrates a 6-axis MEMS sensor and supports sophisticated Real-Time Kinematic (RTK) and Dead-Reckoning (DR) algorithms to fuse the sensor data, GNSS raw data and speed data, etc. to provide sub-meter level positioning accuracy in an open-sky environment
- Support AGNSS
- Embedded flash memory provides the capacity for storing user-specific configurations and future firmware updates.

3 Typical Applications

- High-precision GNSS for Tracking and Positioning
- Navigation where GNSS signal is not available
- 6-axis sensor (3-axis accelerator and 3-axis gyroscope)
- Internet of things

4 GNSS Constellations

The GNSS module is a dual-band GNSS receiver that can receive and track GPS, BeiDou, GLONASS, Galileo, QZSS, IRNSS signals.

4.1 GPS

The module is designed to receive and track GPS L1 C/A signals (1574.397–1576.443 MHz) and L5 signals (1166.22–1186.68 MHz) provided by GPS.

4.2 BeiDou

The module is designed to receive and track BeiDou B1I (1559.052–1563.144 MHz) and B2a (1155.99–1196.91 MHz). The ability to receive and track BeiDou signals in conjunction with GPS results in higher coverage, improved reliability, and better accuracy.

4.3 GLONASS

The module is designed to receive and track GLONASS L1 signals (1597.781–1605.656 MHz) provided by GLONASS.

4.4 Galileo

The module is designed to receive and track Galileo E1 (1573.374–1577.466 MHz) and E5a (1166.22–1186.68 MHz) signals provided by Galileo.

4.5 IRNSS

The Indian Regional Navigation Satellite System (IRNSS) or NavIC is a regional navigation satellite system that transmits additional L5 signals for complying with the requirements of an independent accurate positioning system for users in India. The GNSS module is designed to receive and track IRNSS L5 signals (1175.427–1177.473 MHz) from IRNSS satellites.

4.6 QZSS

The Quasi-Zenith Satellite System (QZSS) is a regional navigation satellite system that transmits additional GPS L1 C/A signals for the Pacific region covering Japan and Australia. The GNSS module can detect and track these signals concurrently with GPS signals, resulting in better availability especially under challenging conditions, e.g., in urban canyons.

5 Augmentation System

5.1 SBAS

The GNSS module supports SBAS (Satellite-Based Augmentation System) broadcast signal reception, and GPS data are complemented by additional regional or wide area GPS enhancement data. The system enhances the data through satellite broadcasting, and this information can be used in GNSS receivers to improve the accuracy of the results. SBAS satellites can also be used as additional signals for ranging or distance measurement, further improving availability. Supported SBAS systems: WAAS, EGNOS, MSAS and GAGAN.

5.2 AGNSS

The module supports AGNSS feature that significantly reduces the module's TTFF, especially under lower signal conditions. To implement AGNSS feature, the module should get the assistance data including the current time, rough position, and LTO data.

5.3 Real-Time Kinematic (RTK)

The GNSS module modules support RTK functionality as rovers. Before supporting the RTK navigation technique, the module needs to receive the RTK correction messages via its UART port. RTK correction messages can be delivered either using a cellular module or other terrestrial network technology. In default configuration, the module will attempt to achieve the best positioning accuracy based on the correction data that it receives. When the module receives an input stream of RTCM messages, it enters RTK float mode. Once it fixes carrier phase ambiguities, the module enters the RTK fixed mode.

The module may be expected to achieve sub-meter level accuracy only when it is in RTK fixed mode. If the module loses the carrier phase lock, at the minimum semaphore required to maintain the RTK fixed mode, it returns to the RTK float mode. The module will also continue to try to resolve carrier phase ambiguity and return to RTK fixed mode after restoring the minimum semaphore.

The current mode of operation is set by the associated NMEA messages.

5.4 Odometer (ODO)

The FWD hardware input is used to input vehicle forward/backward status signals. When it is at low voltage level, the vehicle is moving forward, and when it is at high level, it is moving backward.

The WHEELTICK hardware input is used to input odometer signals from a vehicle. It can be sampled from the wheel revolution sensors or the transmission of the vehicle.

Only cars need to be connected to the FWD signal, electric bicycles do not.

5.5 Geofencing

The GNSS module supports geofence areas, defined on the Earth's surface using a 2D model.

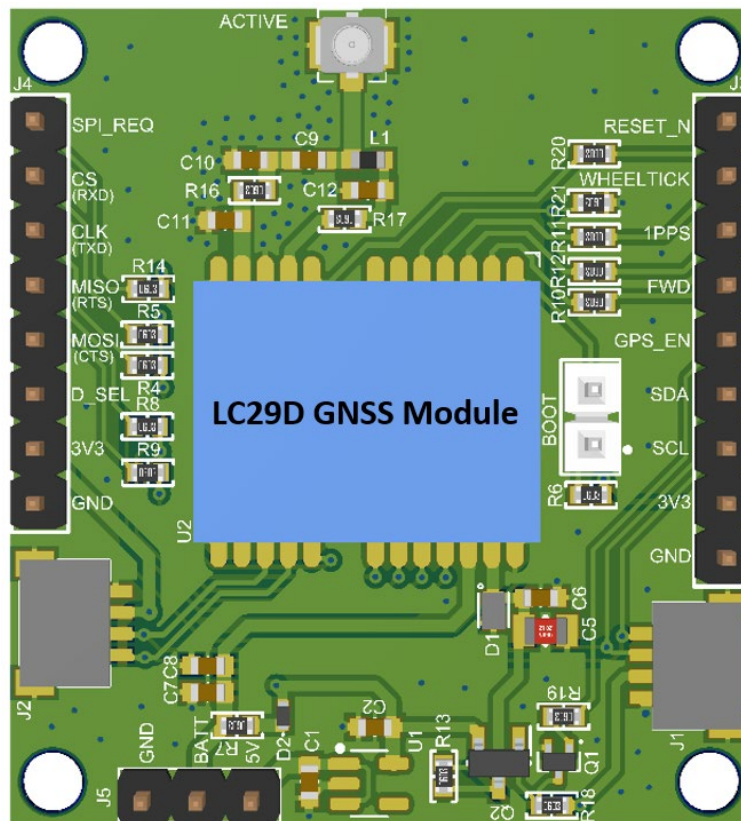
Geofencing is active when at least one geofence area is defined. The current status can be found by polling the receiver. The receiver evaluates whether the current location of each region is within that region or not and signals its status via GEOFENCE pin. Geofencing feature can be configured using geofence messages. The evaluation is activated whenever one or more geofences are configured.

5.6 Dead Reckoning Function

Dead Reckoning is the process of estimating the module's current position based on the last position obtained from GNSS, speed, heading sensor data, etc. With this combined sensor inputs, the system plots the navigation trace when the satellite signals are partially or completely blocked while satellite signals provide updates and correction for sensor drift. With this technology, the system obtains continuous and high-accuracy positioning in environments such as tunnels and urban canyons.

In addition, the module supports output of sensor raw data through UART to support your applications such as driving behavior analysis.

6 Hardware Board and Case Diagram



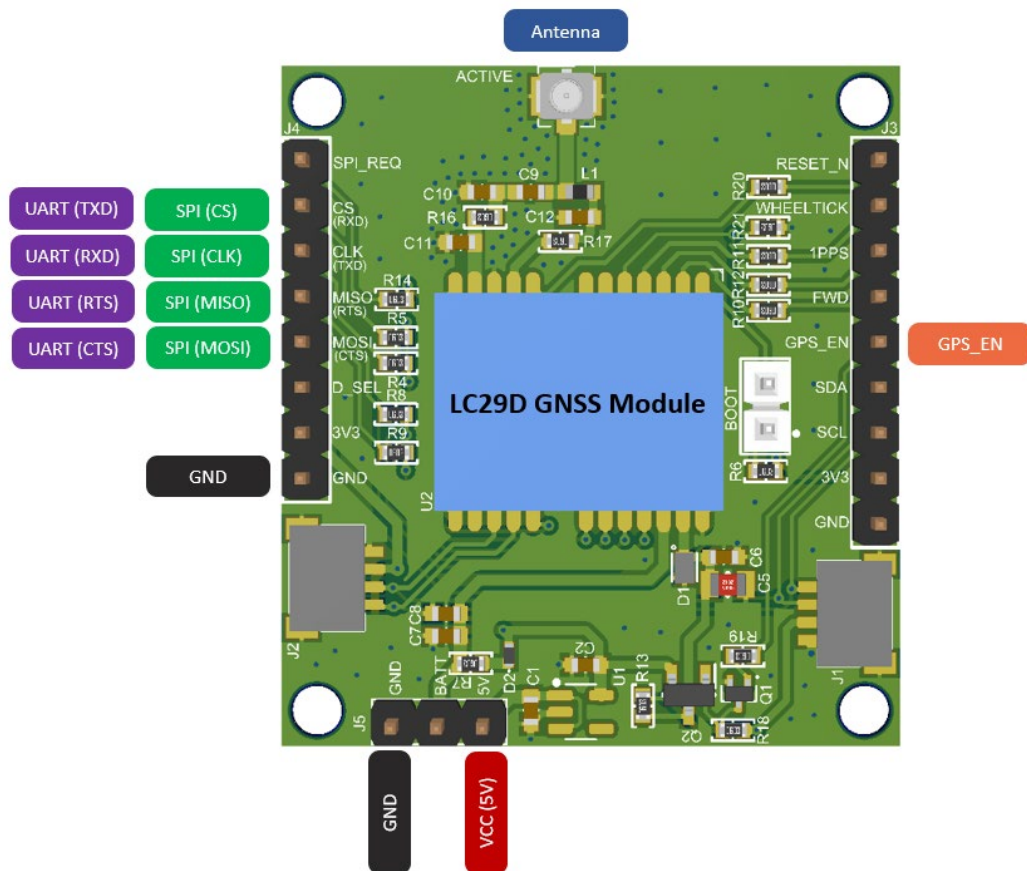
Pin #	Feature	Direction	Description
J3-1	RESET_N	Input	Resets GNSS module
J3-2	WHEELTICK	Input	Odometer wheel-tick input
J3-3	1PPS	Output	One pulse per second
J3-4	FWD	Input	Forward/Backward direction
J3-5	GPS_EN	Input	Enable the GNSS module
J3-6	SDA	Input/Output	I2C data. Currently not used
J3-7	SCL	Input	I2C clock. Currently not used
J3-8	3.3V	Output	Output 3.3V
J3-9	GND	Output	Ground
J4-1	SPI_REQ	Output	Indicates SPI data validity

J4-2	CS (RXD)	Input	SPI Chip-Select or UART Receive data
J4-3	CLK (TXD)	Input/Output	SPI Clock or UART Transmit data
J4-4	MISO (RTS)	Output	SPI master in slave out or UART Request to send
J4-5	MOSI (CTS)	Output	SPI slave output master in or UART Clear to Send
J4-6	D_SEL	Input	Selects SPI or UART interface
J4-7	3.3V	Output	Output 3.3V
J4-8	GND	Output	Ground
J5-1	GND	Output	Ground
2	BATT	Input	Back-up battery
3	5V	Input	Input voltage source (5V)

7 Connecting using the UART or SPI

The GNSS board can be interfaced with either UART or SPI. Connect the board to 5V and GND and then either connect UART or SPI interface to it. Remember to enable GPS_EN by connecting it to a pull-up voltage of 3.3V. The pins used are shown below.

Connect an Antenna suitable for GNSS receivers to the Antenna U.FL connector.



You can use the GNSS board with any device that has UART (Tx/Rx) interface such as an Arduino board or any other hardware board available.

You can also use and connect the GNSS board to a USB port in the computer through a USB-to-UART cable. You can use cable such as this one:

[USB TO UART TTL \(Wires\) Serial Cable \(PL2303HX\) MCP00102W Programmer Arduino Compatible in Elecrow bazaar!](#)

Such a cable has four wires colored as follow:

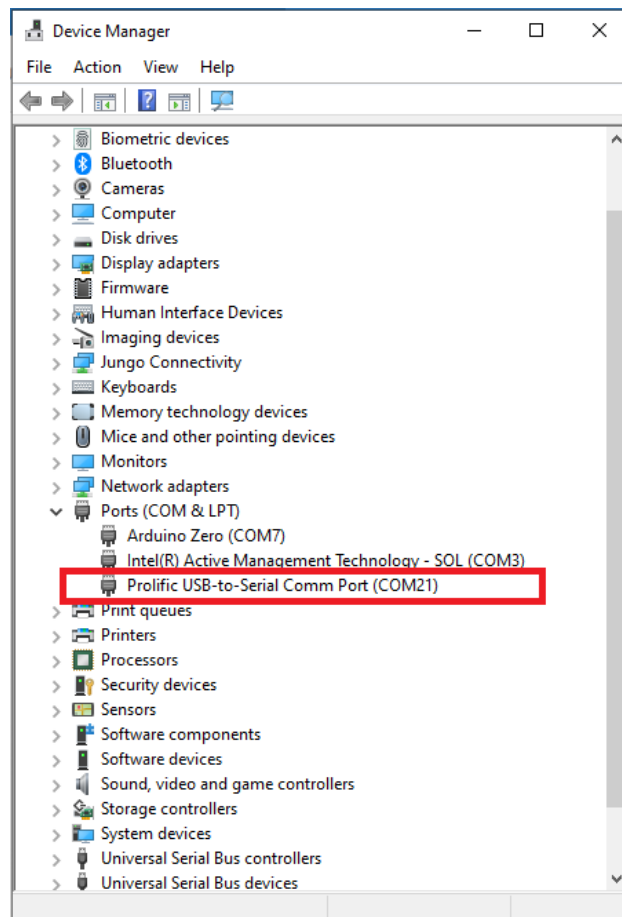
Red: +5V (**Do not Connect to the GNSS board**)

White: TXD

Green: RXD

Black: GND

Connect the **White** cable to **UART(TXD)** and **Green** cable to **UART(RXD)**. Also connect the **Black** cable to the **GND**. Make sure when you connect the USB-to-UART cable, it shows correctly in Windows device manager and all its driver is installed as in this screenshot.

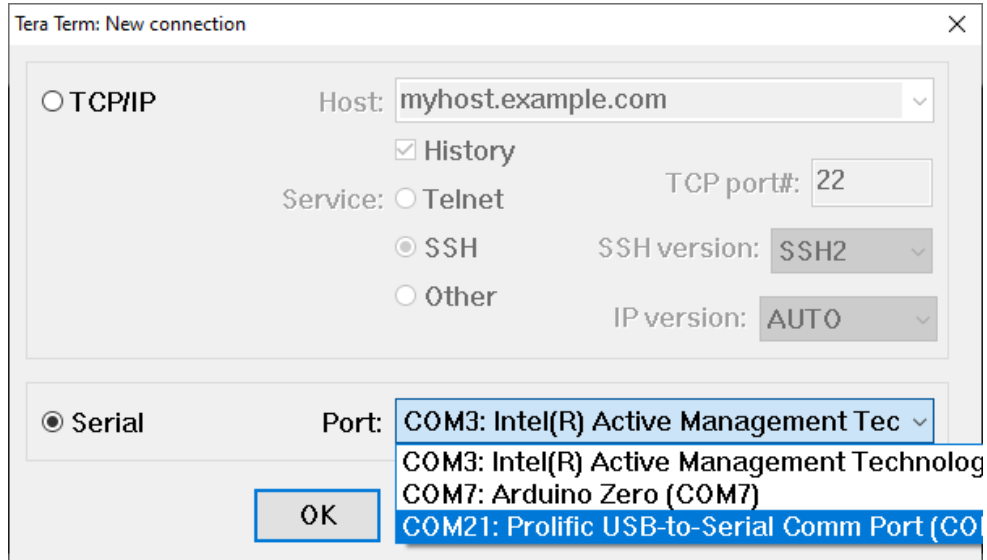


8 Using Serial Terminal

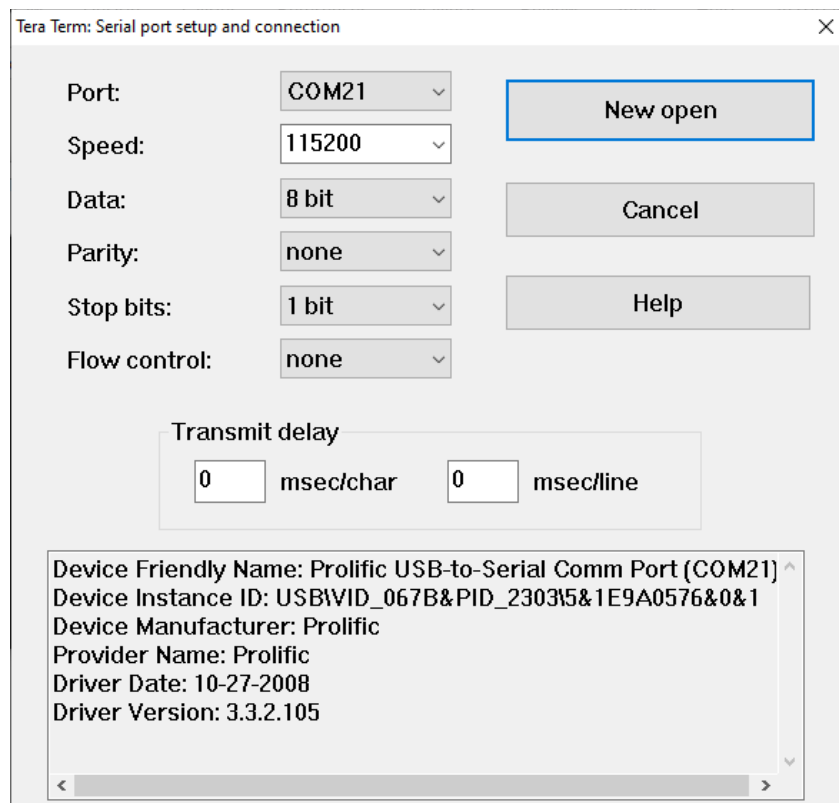
You can use any serial monitor such as the Tera Term tool as the serial terminal. You can download it from here:

<https://osdn.net/projects/ttssh2/downloads/54081/teraterm-4.72.exe/>

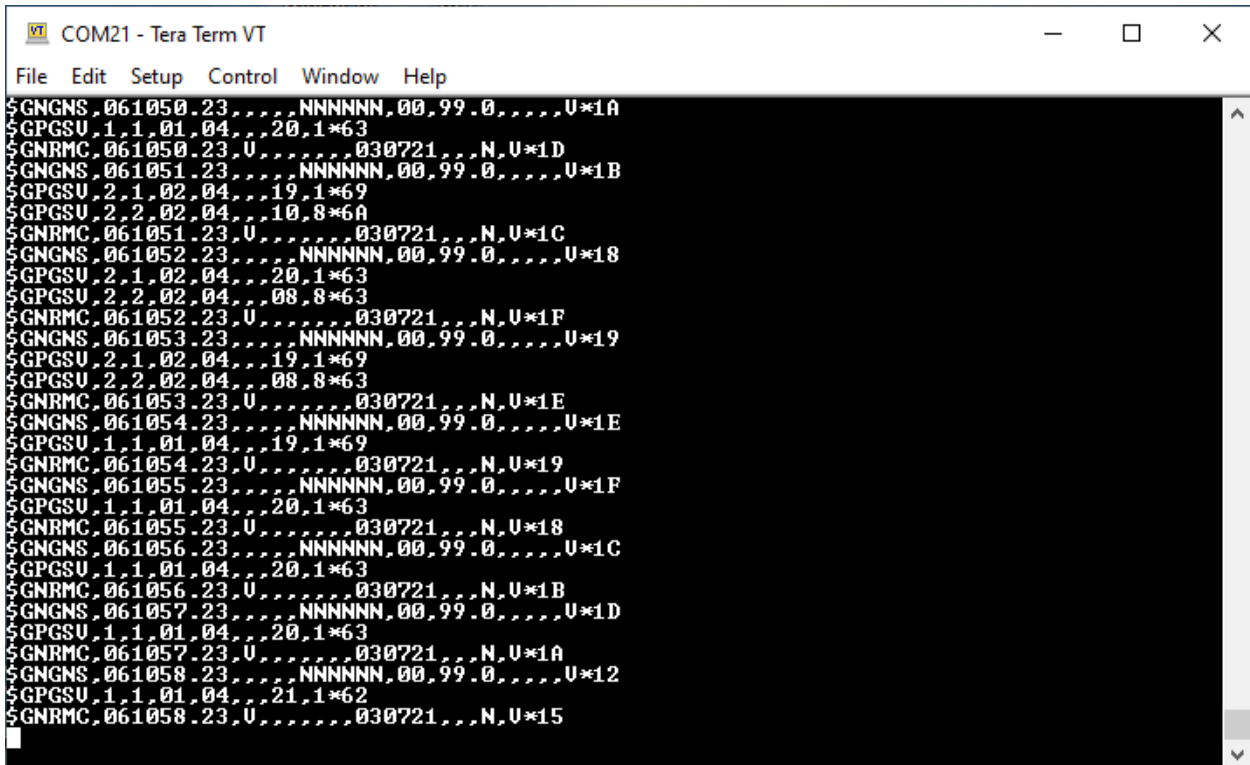
Launch Tera Terminal and select the **Serial** option and select USB-to-Serial port.



In Tera Term, choose **Setup->Serial port** and configure serial ports according to the following:



And you will start to see the GNSS NMEA messages in the terminal Window.

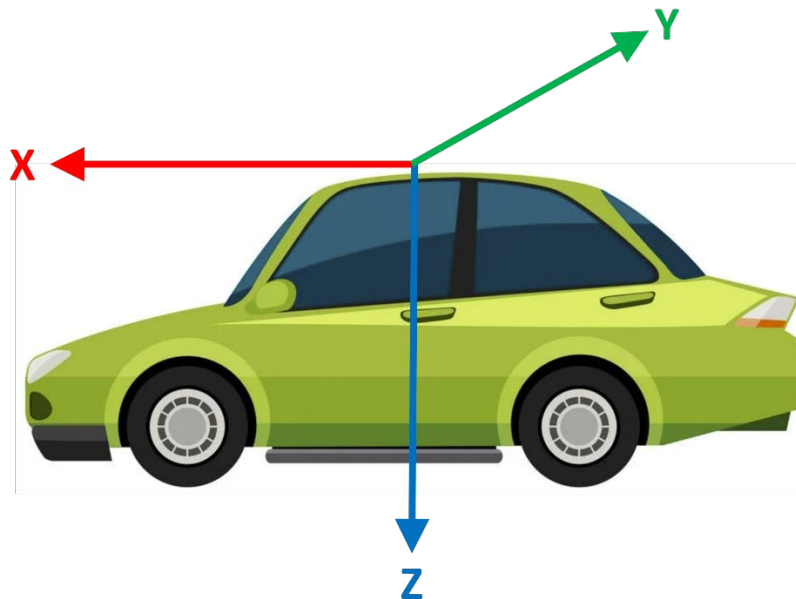


```
COM21 - Tera Term VT
File Edit Setup Control Window Help
$GNGNS,061050.23,,,,,NNNNNN,00,99.0,,,,,U*1A
$GPGSU,1,1,01,04,,20,1*63
$GNRMC,061050.23,U,,,,,030721,,,N,U*1D
$GNGNS,061051.23,,,,,NNNNNN,00,99.0,,,,,U*1B
$GPGSU,2,1,02,04,,19,1*69
$GPGSU,2,2,02,04,,10,8*6A
$GNRMC,061051.23,U,,,,,030721,,,N,U*1C
$GNGNS,061052.23,,,,,NNNNNN,00,99.0,,,,,U*18
$GPGSU,2,1,02,04,,20,1*63
$GPGSU,2,2,02,04,,08,8*63
$GNRMC,061052.23,U,,,,,030721,,,N,U*1F
$GNGNS,061053.23,,,,,NNNNNN,00,99.0,,,,,U*19
$GPGSU,2,1,02,04,,19,1*69
$GPGSU,2,2,02,04,,08,8*63
$GNRMC,061053.23,U,,,,,030721,,,N,U*1E
$GNGNS,061054.23,,,,,NNNNNN,00,99.0,,,,,U*1E
$GPGSU,1,1,01,04,,19,1*69
$GNRMC,061054.23,U,,,,,030721,,,N,U*19
$GNGNS,061055.23,,,,,NNNNNN,00,99.0,,,,,U*1F
$GPGSU,1,1,01,04,,20,1*63
$GNRMC,061055.23,U,,,,,030721,,,N,U*18
$GNGNS,061056.23,,,,,NNNNNN,00,99.0,,,,,U*1C
$GPGSU,1,1,01,04,,20,1*63
$GNRMC,061056.23,U,,,,,030721,,,N,U*1B
$GNGNS,061057.23,,,,,NNNNNN,00,99.0,,,,,U*1D
$GPGSU,1,1,01,04,,20,1*63
$GNRMC,061057.23,U,,,,,030721,,,N,U*1A
$GNGNS,061058.23,,,,,NNNNNN,00,99.0,,,,,U*12
$GPGSU,1,1,01,04,,21,1*62
$GNRMC,061058.23,U,,,,,030721,,,N,U*15
```

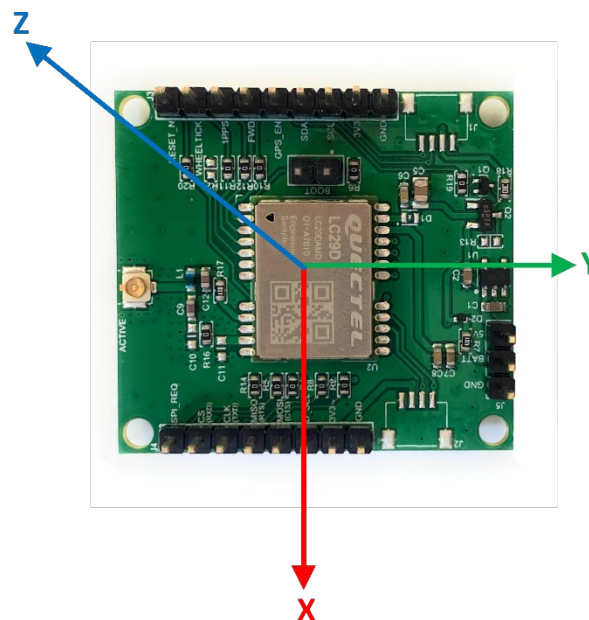
9 DR Configuration

9.1 Setting the Orientation

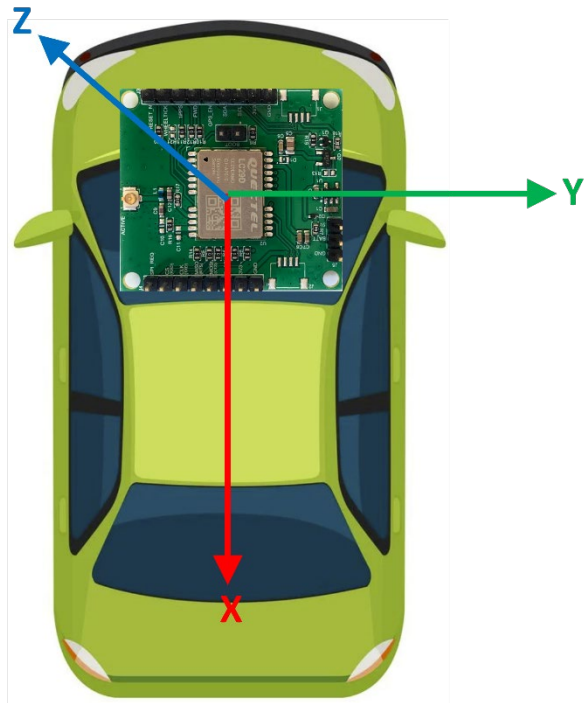
The reference frame axes definitions are shown below. The X axis is the vehicle forward direction, the Y axis is the right side of the vehicle, and the Z axis is the downward direction.



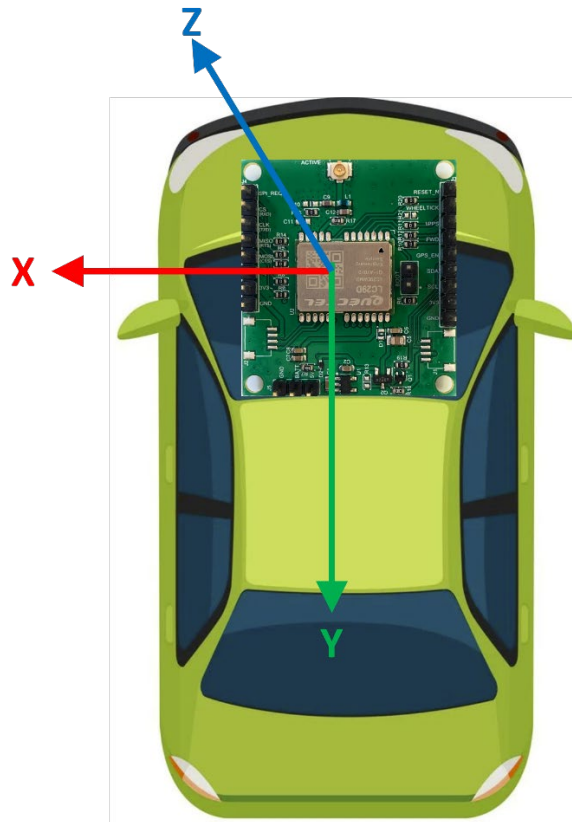
The orientation of the module is shown below:



Below show some examples for board installation:



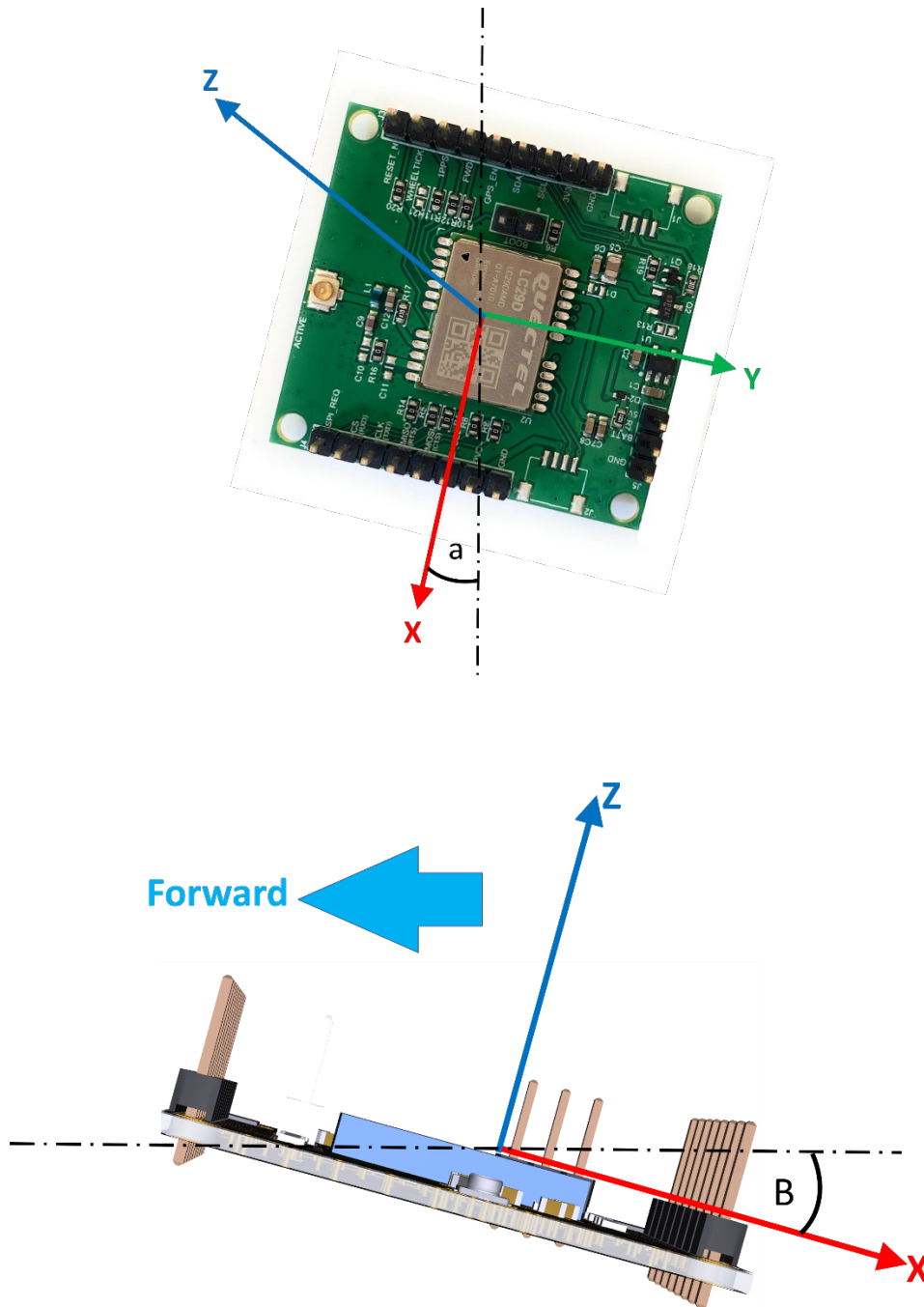
if the board installation direction like the above picture, the X axis is inverted with reference, the Y axis is same with the reference frame, and the Z axis is also inverted with the reference frame, so the configuration should be **-X Y -Z**, the command is **\$PQTMCFGORIENTATION,1,-X Y -Z*66**.



If board installation direction like the above picture, the X axis is inverted with Y axis of reference frame, the Y axis is inverted with the X axis of reference frame, and the Z axis is also inverted with the Z axis of reference frame, so the configuration should be **-Y -X -Z**, the command is **\$PQ TMCFGORIENTATION,1,-Y -X -Z*4B**.

10 Mounting

When mounting the LC29D on the carrier, need to keep the yaw, pitch and roll angle within 5 degrees to the reference frame.



In the real mounting, need to make sure that $-5^\circ \leq a \leq 5^\circ$, $5^\circ \leq B \leq 5^\circ$.

11 Dead-Reckoning Calibration

- 1) Fix the module on the vehicle frame. Any displacement, turn or tilt of the device, although small, will cause performance issues and/or void the calibration process making it fail.
- 2) The calibration process should be performed on good GNSS signal conditions and clear sky view.
- 3) Power up the module then start the vehicle on in a plain surface and keep it still for at least 30 seconds.
- 4) Start driving the vehicle in good GNSS signal conditions. The module will start the self calibration process which would be completed in a few minutes.
- 5) The calibration process ends when the <SolType> of \$PQTMINS message indicates combined solution(GNSS + DR).

12 Enable Dead-Reckoning and 6-axis Sensor

You can enable or disable the GNSS and the raw data from the internal 6-axis sensor and for dead-reckoning. Use the following command to enable or disable GNSS and/or sensor.

```
$PQTMCFGEINMSG,<RW>,<INS>,<IMU>,<GPS>,<Rate>*<Checksum><CR><LF>
```

If it succeeds, it will output:

```
$PQTMCFGEINMSGOK*16
```

After issuing the above command, it has to be saved using the following command

```
$PQTMSAVEPAR*5A
```

Table 1: Config GNSS and sensor parameters.

Field	Unit	Meaning
<RW>		0 = Read current configuration. 1 = Config.
<INS>		Enable/disable PQTMINS message. 0 = Disable. 1 = Enable.
<IMU>		Enable/disable PQTMIMU message. 0 = Disable. 1 = Enable.
<GPS>		Enable/disable PQTMGPS message. 0 = Disable. 1 = Enable.
<Rate>		set the output rate of PQTMINS and PQTMIMU messages. It can be 10, 20, 50, 100. When set to 100, the maximum output rate is dependent on baud rate.

Issue the following two commands and then restart the board (power it OFF and then ON)

Command: **\$PQTMCFGEINSMMSG,1,1,1,1,10*3F**
response: **\$PQTMCFGEINSMMSGOK*16**

Command: **\$PQTMSAVEPAR*5A**
Response: **\$PQTMSAVEPAROK*5E**

The following read current configuration:

Command: **\$PQTMCFGEINSMMSG,0*0E**
Response: **\$PQTMINSMSG,0,1,1,1,10*7C**

The following can be done through QCOM tool or any serial terminal connected to the board. Enter each command above and click **Send Command**.

The screenshot displays the QCOM_V1.6 software interface. The 'COM Port Setting' section is configured with COM Port 3, Baudrate 115200, StopBits 1, Parity None, ByteSize 8, and Flow Control No Ctrl Flow. The 'Command List' section contains 29 commands, with the first 11 checked. The 'Operation' section shows the 'Input String' field containing '\$PQTMSAVEPAR*5A' and the 'Send Command' button highlighted with a red box. The 'Run Times' is set to 10 and 'Delay Time(mS)' is set to 1000.

Choose All Commands	HEX	Enter	Delay(mS)
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	2
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	3
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	4
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	5
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	6
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	7
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	8
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	9
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	10
<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	11
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	12
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	13
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	14
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	15
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	16
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	17
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	18
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	19
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	20
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	21
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	22
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	23
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	24
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	25
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	26
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	27
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	28
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	29

13 Sensor Messages

The GNSS receiver emits this message. The message contains the IMU Raw Data: Acceleration, Angular Rate and Hardware Wheel Ticks.

```
$PQTMIMU,<Timestamp>,<ACC_X>,<ACC_Y>,<ACC_Z>,<AngRate_X>,<AngRate_Y>,<AngRate_Z>,<TickCount>,<LastTick_Timestamp>*<Checksum><CR><LF>
```

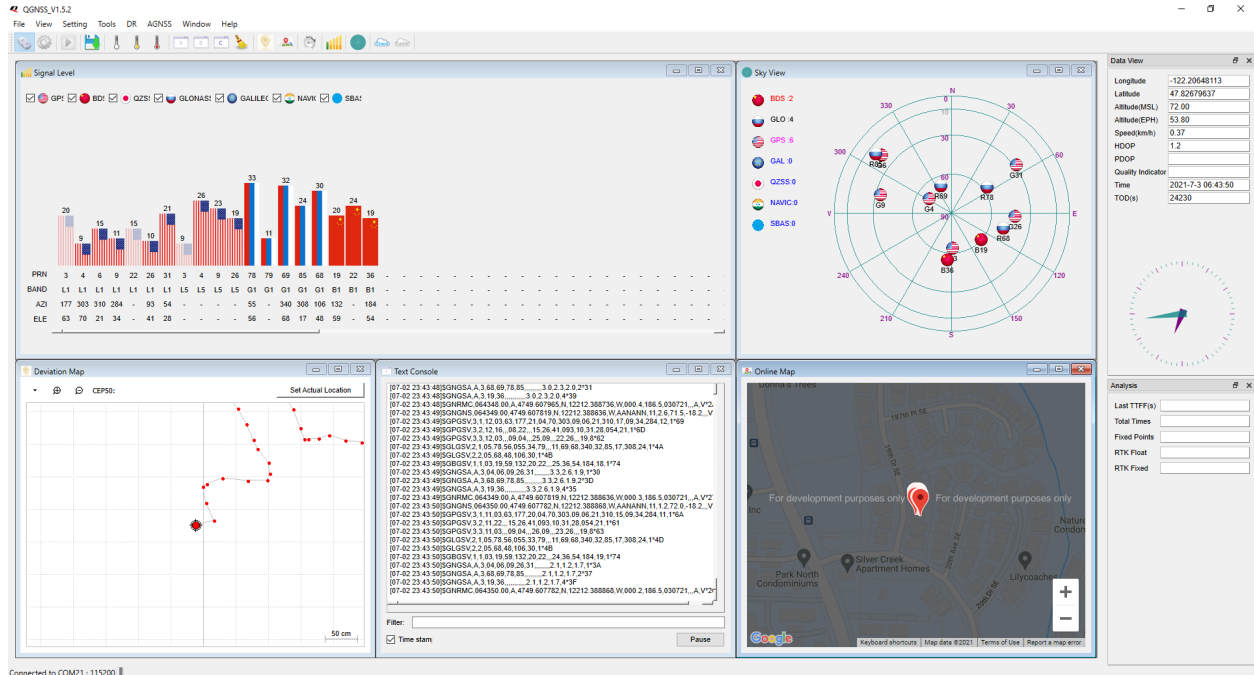
Table 2: PQTMIMU message parameters.

Field	Format	Unit	Meaning
<Timestamp>	Numeric	ms	Milliseconds since turn on. 32-bit unsigned integer
<ACC_X>	Numeric	G	X-Acceleration
<ACC_Y>	Numeric	G	Y-Acceleration
<ACC_Z>	Numeric	G	Z-Acceleration
<AngRate_X>	Numeric	deg/s	X-Angular Rate
<AngRate_Y>	Numeric	deg/s	Y-Angular Rate
<AngRate_Z>	Numeric	deg/s	Z-Angular Rate

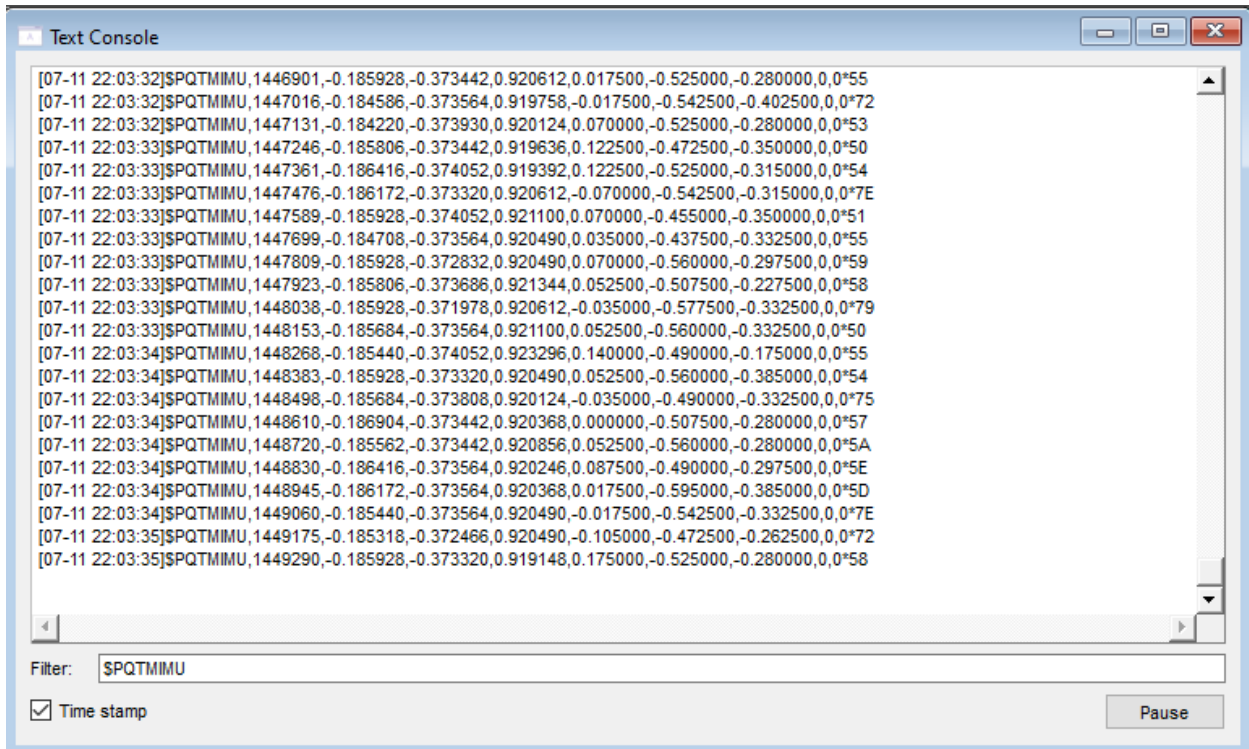
14 Using QGNSS Tool

QGNSS tool is a visual tool where you can see different information about GNSS position including satellites acquired, GNSS NMEA sentences, online Google map, Sky View, Deviation maps and Data View.

Connect the QGNSS to the COM port and click **File->Connect** and you will be able to see a lot of visual data.



You can view the sensor messages only in the GNSS tool by filtering those messages only as in this screen-shoot:



15 Using Arduino IDE

Download the Arduino sketch for the LC29D module from the GitHub and upload it to the NB-IoT board. Connect the UART of the GNSS board to the UART of the NB-IoT board. You shall see the GNSS output messages on Arduino serial monitor.

```
COM7
$PQTMINS,66360,0,,,,,1.847466,-17.606845,-10.124997*7B
$PQTMIMU,66466,0.033672,0.302072,0.957700,0.192500,-0.052500,-1.032500,0,0*52
$PQTMINS,66468,0,,,,,1.861634,-17.615494,-10.123835*79
$PQTMIMU,66576,0.030622,0.306220,0.959286,0.017500,-0.647500,-0.210000,0,0*58
$PQTMINS,66578,0,,,,,1.847838,-17.619227,-10.121664*75
$GNXDR,000105.00,,,,,MNNNNN,00,99.0,,,,,V*1D
$GPGSV,1,1,01,31,,,23,1*66
$GNRMC,000105.00,V,,,,,010712,,,N,V*18
$PQTMIMU,66683,0.031720,0.305244,0.956846,0.052500,-0.332500,-0.455000,0,0*5A
$PQTMINS,66685,0,,,,,1.847962,-17.622244,-10.121664*77
$PQTMIMU,66798,0.031476,0.306708,0.957822,0.052500,-0.595000,-0.367500,0,0*51
$PQTMINS,66800,0,,,,,1.847796,-17.624943,-10.121664*7B
$PQTMIMU,66913,0.029646,0.306220,0.957700,0.035000,-0.577500,-0.385000,0,0*50
$PQTMINS,66915,0,,,,,1.846085,-17.628883,-10.121664*7C
$PQTMIMU,67028,0.030134,0.305366,0.966728,-0.507500,1.365000,-1.522500,0,0*5D
$PQTMINS,67031,0,,,,,1.795576,-17.620580,-10.091875*7F
$PQTMIMU,67148,0.032818,0.304878,0.958310,0.035000,-0.682500,-0.332500,0,0*57
$PQTMINS,67151,0,,,,,1.848445,-17.635828,-10.117703*7C
$PQTMIMU,67264,0.029158,0.305732,0.958188,0.017500,-0.542500,-0.315000,0,0*54
$PQTMINS,67266,0,,,,,1.846952,-17.639003,-10.117703*73
$PQTMIMU,67378,0.031598,0.305854,0.957456,0.087500,-0.595000,-0.280000,0,0*54
$PQTMINS,67380,0,,,,,1.847173,-17.642341,-10.117703*79
$PQTMIMU,67483,0.030988,0.305366,0.957090,0.087500,-0.595000,-0.262500,0,0*56
$PQTMINS,67485,0,,,,,1.847296,-17.644114,-10.117703*77
$PQTMIMU,67589,0.030988,0.305854,0.958188,0.052500,-0.455000,-0.367500,0,0*51
$PQTMINS,67591,0,,,,,1.846211,-17.645589,-10.117703*7C
$GNXDR,000106.00,,,,,MNNNNN,00,99.0,,,,,V*1E
$GPGSV,1,1,01,31,,,24,1*61
$GNRMC,000106.00,V,,,,,010712,,,N,V*1B
$PQTMIMU,67698,0.030500,0.306586,0.956968,0.017500,-0.402500,-0.402500,0,0*55
$PQTMINS,67700,0,,,,,1.846118,-17.649372,-10.117703*72
 Autocroll  Show timestamp
Carriage return 9600 baud Clear output
```