Proceedings of 12th IOE Graduate Conference
Peer Reviewed
ISSN: 2350-8914 (Online), 2350-8906 (Print)
Year: 2022 Month: October Volume: 12

Development of a Low-cost GNSS Data Logger and Deployment in Dynamic mode for GNSS Data Logging and Analysis

Ajay Kumar Thapa [a], Dinesh Manandhar [b], Krishna Prasad Bhandari [c]

[a] *Department of Geomatics Engineering, Pashchimanchal Campus,IOE,Tribhuvan University,Nepal*
[b] *Center for Spatial Information Science,The University of Tokyo,Japan*
[c] *Center for Space Science and Geomatics Studies,Pashchimanchal Campus,IOE,Tribhuvan University,Nepal*
[a] ajaythapa0061@gmail.com , [b] dinesh@csis.u-tokyo.ac.jp , [c] bhandarikrishna@wrc.edu.np

Abstract

The global network of satellite constellations known as GNSS is used to determine location and time worldwide using a constellation of many artificial satellites, ensuring that at least four satellites are visible at once from any spot on Earth. A GNSS receiver uses the signals it has received from the visible satellites to calculate the transit time and the distance to each satellite. Low-cost GNSS receivers are those receivers with low costs and compact designs that are frequently utilized for determining position, velocity and time(PVT). The NEO-M8T receiver module is a single frequency, multi-GNSS receiver when paired with a GNSS patch antenna, it can offer a significant level of positional accuracy. A Low-cost GNSS Data Logger using inexpensive hardware and freely available software which is portable, and operable in static and dynamic mode was developed capable of obtaining PVT data as well as raw GNSS data. The PVT information was obtained for both city commute as well as the highway, and visualization was performed between time and speed. In both of the cases, the variation of speed with time, traffic and road conditions could be seen from the comparison. On further analysis, it was found that for the city commute, the decline in speed was seen mainly at the road intersections depicting increase in traffic volume and a necessity to decelerate the vehicle.Whereas in highway, the decline in speed was seen along the road sections where the road conditions are bad or in off-road sections. The raw GNSS data was also automatically logged and with post-processing techniques the positioning could be further enhanced. Overall,this study further recommends studies related to speed monitoring,driving behavior and traffic congestion with the application of low-cost GNSS receivers.

Keywords

low-cost GNSS receiver,ublox NEO-M8T, Internet of Things (IoT), low-cost GNSS Data Logger, Traffic analysis

1. Introduction

The formal definition of the Global Navigation Satellite System (GNSS) is the global system for determining location and time that consists of one or more constellations of satellites [1, 2]. The Global Positioning System widely known as GPS and GLONASS which stands for Globalnaya Navigatsionnaya Sputnikovaya Sistema, were once the two main constellations that made up the GNSS system. In contrast, the Galileo constellation in Europe and the Compass systems, also known as BeiDou, in China, have both been developed and implemented. The present state of GLONASS has been rejuvenated and is completely functioning, while GPS is currently running at its full capacity. Additionally, modernization of GPS and GLONASS is ongoing [3]. In addition, a number of satellite-based augmentation systems (SBAS), including regional satellite systems like Quasi-Zenith Satellite System(QZSS) of Japan, and Indian Regional Navigation Satellite System (IRNSS) of India have been installed and are being continuously upgrading.

The newly evolved systems along with the modernization of GPS and GLONASS, are having a substantial impact on the satellite navigation industry. A better and broader variety of Position Navigation and Time (PNT) applications may be made possible by the next generation of GNSS. Since the dual-frequency system GPS was all that was available in the past, there are already a lot more satellites visible now that are delivering navigation data at multiple frequencies. If many satellite systems are employed, the accuracy, consistency, and accessibility of precise location will be considerably increased when compared to GPS-only solutions [4, 5]. Additionally, this will enable real-time kinematic applications to significantly reduce the startup time.

GNSS uses a constellation of several artificial satellites to pinpoint the location of a receiver on land, oceans, or in space. Latitude, longitude, and altitude are used to estimate the receiver location using computed distances to multiple satellites. A navigation message is broadcast continually by each satellite. The receiver utilizes the messages it has received from visible satellites to calculate its transit time and the distance to each one. A system with a sufficient satellites must be deployed for each constellation in order to ensure that at least 4 satellites are visible at once from any location on Earth. The affordability, portability, ease of use, and wide variety of applications of low cost GNSS receivers are now driving up their use and popularity. Due to their low cost and compact design, these receivers are frequently utilized in IoT(Internet of Things) projects, geospatial applications, and monitoring and determining location and velocity. These inexpensive receivers can operate with a single frequency or multiple frequencies, with the latter offering more accuracy.

The single frequency, multi-GNSS UBlox NEO-M8T receiver module is small, portable, and reasonably priced. It can provide a moderate level of location accuracy when used in conjunction with a tiny GNSS antenna [6]. The receiver is small, reasonably priced, and readily available. It can also provide location data in NMEA format. In a static environment, the DGNSS positioning accuracy of the single-frequency NEO-M8T receiver with a patch antenna is submeter level, while in a dynamic environment, it is capable of meter level [7]. The positional accuracy of a combination of multiple-GNSS system is higher compared to a stand-alone GNSS constellation and the NEO-M8T receiver serves as a lightweight, reasonably priced multi-GNSS capable sensor with Real-Time Kinematics (RTK) capabilities [8], providing real-time geo-referencing for quick, dependable real-time dynamic data processing. The capability of the NEO-M8T module to support multiple GNSS constellations enables it for comparative study of single and multi-GNSS positioning accuracy in varying operating environment situations. According to [9] the accuracy of the multi-GNSS system such as BeiDou/GPS unified positioning was significantly better than that of the GPS only positioning method in X, Y, Z directions for NEO-M8T receiver.

A comprehensive description can be found in [10] from which it is evident that low-cost single frequency receivers like the NEO-M8T are also viable for GNSS-Reflectometry (GNSS-R) applications,where a single antenna capable of acquiring data about the interference between direct signals and reflected signals in static mode application. The construction of a dependable mobile mapping system using low-cost receivers like the M8T and technologies like Ultra-Wideband demonstrates how these systems can operate in tandem with other devices and achieve positive outcomes [11]. As indicated previously, [12] used a series of practical observations taken to show how RTK positioning of the NEO-M8T is used in precision farming and agriculture where the module acted as a portable, cost-effective, and energy-efficient tool. The NEO-M8T sensor validates how these low-cost receivers can be used to accurately monitor and record the motion of moving objects as demonstrated in [13] where pedestrian behaviors like velocity, acceleration, and deceleration were recorded using the receiver and studied. As demonstrated in [14], a low-cost GNSS module used for speed monitoring provided similar results to the vehicle's speedometer and used for identifications and quantification of road sections where speeding occurs. When contrasted to a more capable ZED-F9P, having 178 channels and can receive dual frequency data from all GNSS satellites, the NEO-M8T, a 72-channel module that can track all four current GNSS, performed satisfactorily for the time transfer tests [15].

All of the research demonstrates that the NEO-M8T receiver module is a dependable tool for collecting GNSS data and performing GNSS research. This low-cost GNSS receiver may be used to collect GNSS data when paired with inexpensive mobile computers like the Raspberry Pi. When the NEO-M8T module and Raspberry Pi are used together, a low-cost GNSS data logger that is portable, inexpensive, and has a wide range of possible applications is produced. Also, at the present, the process of logging GNSS data is mainly carried out via CORS(Continuously Operating Reference Stations), Differential GPS, and Handheld GPS. The data received from most of the existing handheld GPS devices in the market may give meter-level precise data, but the first two offer solid performance but are expensive, need auxiliary equipment for setup, and are relatively not portable. As a result, a trustworthy, affordable device that is highly customizable for its operation can record raw GNSS data is required. Additionally, this device can provide a selection of options for correction in real-time, post-processing capabilities using the obtained data, and a huge variety of additional GNSS signal uses.

2. Study Area

Pokhara Metropolitan City is the research region under consideration. Pokhara is Nepal's second-biggest city in terms of population and the country's largest metropolitan city in terms of area. It is situated in Nepal's Gandaki Province's Kaski district. It is situated around 200 kilometers (km) west of Kathmandu, Nepal's capital city. Over a geographical area of 464 square kilometers, Pokhara is divided into a total of 33 wards. Geographically it lies in $83^{0}58'30"$ to $84^{0}02'03"$ East Longitude and $28^{0}10'00"$ North to $28^{0}16'00"$ North Latitude. For city Commute data logging, the route started from Nagdhunga, Pokhara-8 to Chorepatan, Pokhara-17 as shown in Figure 1.

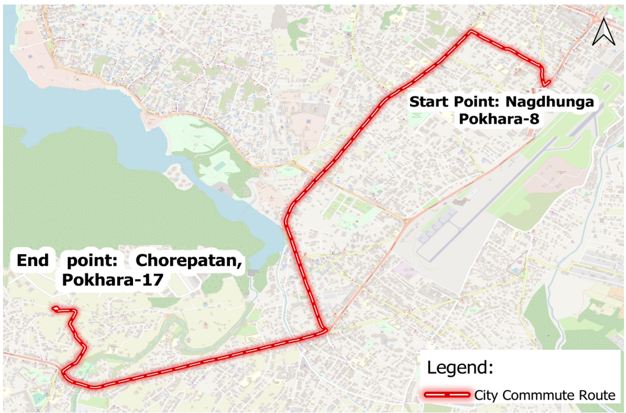

Figure 1: Route for data logging in City Commute

Pokhara-Baglung highway is among the busiest Highway in Nepal that connects the capital city Kathmandu with Pokhara and Baglung. This highway is 72 kilometers commencing from Pokhara -Zero Kilometer to Baglung municipality. For dynamic data logging in highway travel, the route starting from Pashchimanchal Campus, Pokhara-16 to Annapurna Rural Municipality Office, Dhikur Pokhari along the Pokhara-Baglung Highway was selected as illustrated in Figure 2.

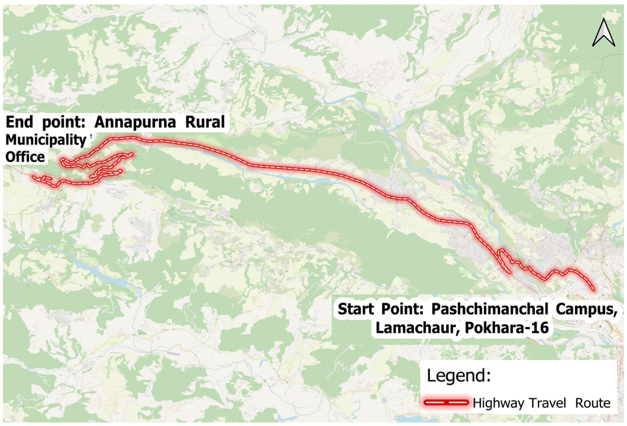

Figure 2: Route for data logging in Highway

3. Research Objective

3.1 General Objective

The General objective of this study is:

• To build a low-cost GNSS data logger utilizing a Raspberry Pi-4B and a NEO-M8T receiver module.

3.2 Specific Objective

The specific objectives of this study are:

• To record GNSS data in dynamic mode: City Commute and Highway Travel using a Low-Cost GNSS Data Logger

• To analyze the dynamic data that the Low-Cost GNSS Data Logger has acquired in Dynamic mode.

4. Methodology

4.1 Architecture of Low-Cost GNSS Data Logger

The methodology used is an interdisciplinary approach incorporating GNSS and IoT technology. The flowchart of the methodology is displayed in Figure 3. The Space segment of the GNSS system,the GNSS satellite constellation transmit the GNSS signals. These signals are received by the patch antenna which communicates with the NEO-M8T single frequency receiver, which was connected to the Raspberry Pi-4B through USB Serial connection.When power is supplied to the Raspberry Pi two data outputs: GNSS raw data and PVT data will be recorded.The recorded data was stored locally on the Raspberry-Pi storage on the SD Card.

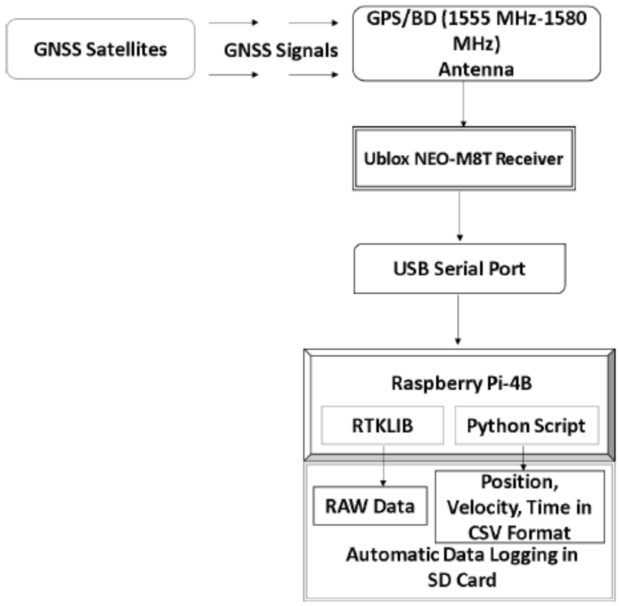

Figure 3: A Flow chart representation of the Low-cost GNSS Data Logger System Architecture

4.2 Device Setup

Establishing the appropriate devices and peripherals is the first stage in creating the Low-cost GNSS Data Logger. The Raspberry Pi 4B model served as the study's mobile computer. First, the official Raspbian OS was downloaded and installed onto the SD Card from the Raspberry Pi website. The installed operating system has the codename "bullseye" as indicated in Figure 4.

```
pi@raspberrypi:~ $ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 11 (bullseye)"
NAME="Raspbian GNU/Linux"
VERSION_ID="11"
VERSION="11 (bullseye)"
VERSION_CODENAME=bullseye
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
pi@raspberrypi:~ $
```

Figure 4: Raspberry Pi-4B Operating System setup

The LCD display configuration was then completed. The Raspberry Pi was coupled with a 3.5-inch LCD display. The appropriate configurations for the display were made when the LCD driver was copied from GitHub. Regular OS backups were performed. It is because the device wouldn't boot up if there were any faults after the system files and scripts were altered. So, the functioning version of the Raspbian OS was backed up. The Raspberry Pi's broadcasted IP address was searched for and discovered. The Raspberry Pi could be accessed from the desktop using it. The LCD display setup was also performed. Then, the Raspberry Pi was also accessed remotely through a computer using the VNC Viewer and its interface is displayed in Figure 5.

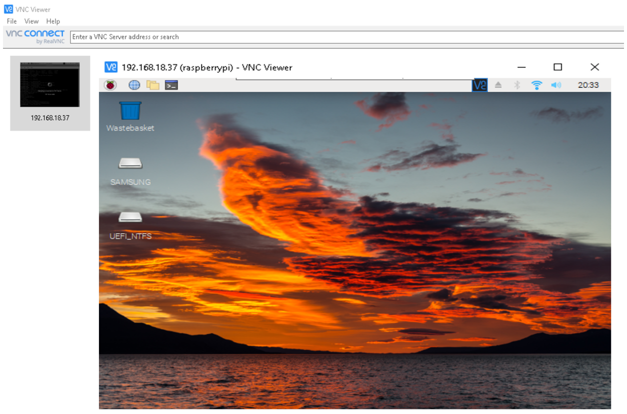

Figure 5: Remote Access to the Raspberry-Pi using desktop software

Additional freely available software was used to connect and transfer files from desktop to the Raspberry Pi. The required files were copied to the Raspberry Pi, and the command line was used to build the RTKLib application in the Raspberry Pi. The final setup of the GNSS Data Logger is shown in Figure 6.

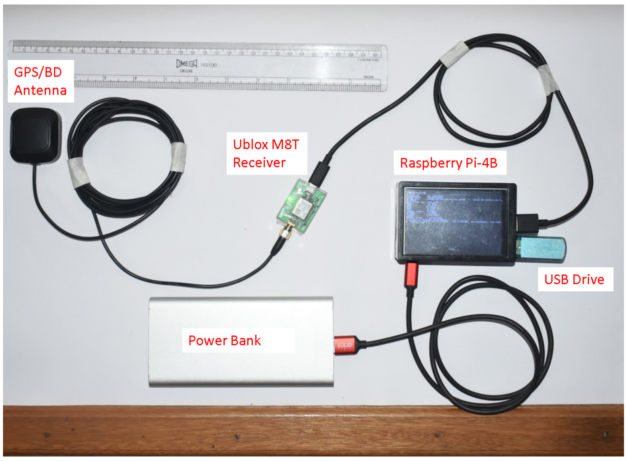

Figure 6: GNSS Data Logger setup

4.3 Data Logging

Utilizing the STR2STR app, GNSS raw data logging was carried out with the proper data logging parameters also being given which is shown in Figure 7. It was attached to the NEO-M8T receiver, and GNSS raw data was logged.

```
                   ���y�sA%�|��A(�����xs=!��xA*�5�A��D(�r�%CeqA9FR2¸A��,��a���b
,               ������@D��
���.ji��@?�i�$GNRMC,145550.00,A,2811.60232,N,08357.45948,E,0.176,,310122,,,D,V*1
0
$GNVTG,,T,,M,0.176,N,0.327,K,D*3E
$GNGGA,145550.00,2811.60232,N,08357.45948,E,2,12,0.79,777.4,M,-40.1,M,,0000*65
�b(������<�����������?¸��q����~$GNGSA,A,3,05,20,24,193,194,14,41,12,19,195,02,06
,1.43,0.79,1.19,1*34
$GNGSA,A,3,31,25,04,19,24,,,,,,,,1.43,0.79,1.19,3*0D
�b!˚�k���/��!��&�$GPGSV,5,1,18,02,65,292,35,05,19,191,48,06,48,027,31,11,72,293,
37,0*6A
$GPGSV,5,2,18,12,44,319,31,14,08,143,39,17,21,081,33,19,36,064,17,0*65
$GPGSV,5,3,18,20,47,177,49,24,22,249,37,28,,,48,40,44,229,,0*51
$GPGSV,5,4,18,41,57,182,47,50,17,105,44,193,17,095,42,194,12,140,40,0*6C
$GPGSV,5,5,18,195,35,069,33,196,35,094,,0*6D
$GAGSV,2,1,08,04,05,151,39,10,,,39,19,08,098,35,24,77,252,35,0*48
$GAGSV,2,2,08,25,28,318,26,31,40,159,45,33,13,048,,36,04,245,,0*79
$GNGLL,2811.60232,N,08357.45948,E,145550.00,A,D*70
$GNZDA,145550.00,31,01,2022,00,00*79
�b ˚�k���/��!��$;�b"˚�k���/��!��(єb����GA�l��sA�<Nn��A��dE��;]����pAA����A��CA
```

Figure 7: GNSS Raw Data Logging in Rapberry Pi using RTKLiib

Additionally, the Raspberry Pi was set up so that when the pen drive was ejected, the pi would shut down and the data logging would stop. A Python program was developed for automatic data logging that filters the raw GNSS data to only get the information that is required, such as longitude, latitude, altitude, vehicle speed, UTC time, and local time of the device as shown in Figure 8. This script will produce a CSV file with this data in it. The logged data was moved to the local computer from the Raspberry Pi storage and locally saved there. The Bash.rc and Crontab scripts were used to automate data logging.

```
 GPS reading
----------------------------------------
latitude     28.1934308
longitude    83.9576954
time utc     2022-03-03T08:50:05.000Z  +  2022-03-03T08:50:05.000Z
altitude (m) 790.483
epx          5.771
epy          5.781
epv          2.042
ept          0.005
eps          0.15
speed (m/s)  0.01
climb        0.008
track        0.0
mode         3
Local Time    14:35:05
------------Data Logging------------
['2022-03-03T08:50:05.000Z', 28.1934308, 83.9576954, 790.483, '14:35:05']
```

Figure 8: Automatic Data PVT Logging using Python Script

The data logging process was automatic and started once power was provided to the Raspberry Pi hardware. The data was logged on two files simultaneously: first for the raw data logged by the RTKLib software, the second file is the .CSV file recorded with the execution of the python script. When the data logging process is completed, the session could be stopped by ejecting a USB drive, which was programmed in such a way that when ejected, the USB drive will trigger the Raspberry pi to shut down.Data was logged for both the commute in the city inside the Pokhara valley was taken from Prithivi Chowk to Chorepatan and on the highway,the route from Pashchimanchal Campus, Lamachaur to Dhikur Pokhari.

4.4 Data Transfer and Visualization

The data was transferred from Raspberry pi to the local computer. GNSS raw data was imported to the local computer, where U-center software was used to access and visualize it.

Additionally, necessary data were directly recorded and saved in CSV format with the use of a Python script. The CSV file was uploaded to Google Earth application where the dynamic data recorded by data logger was displayed and inspected as illustrated in Figure 9.

Figure 9: Data visualization for City commute data in Google Earth Application

5. Results and Discussion

The low-cost GNSS data logger was developed utilizing the Ublox NEO-M8T receiver,GPS/BD Patch antenna, Raspberry pi 4B hardware, RTKLIB software, and Python script. The developed GNSS Data Logger was successfully able to record the data in .UBX format(GNSS raw file format) as well as PVT information in .CSV file format. The GNSS data was logged in two different traffic conditions and during this logging sessions data was logged in both the above mentioned file formats simultaneously. After the completion of the logging session, a USB drive previously programmed and connected to the Raspberry pi was ejected,thus triggering the Raspberry pi to stop the logging session and shutdown automatically. This also prevented the data from being corrupted.

Figure 10: Visualization of City Route in Google Earth

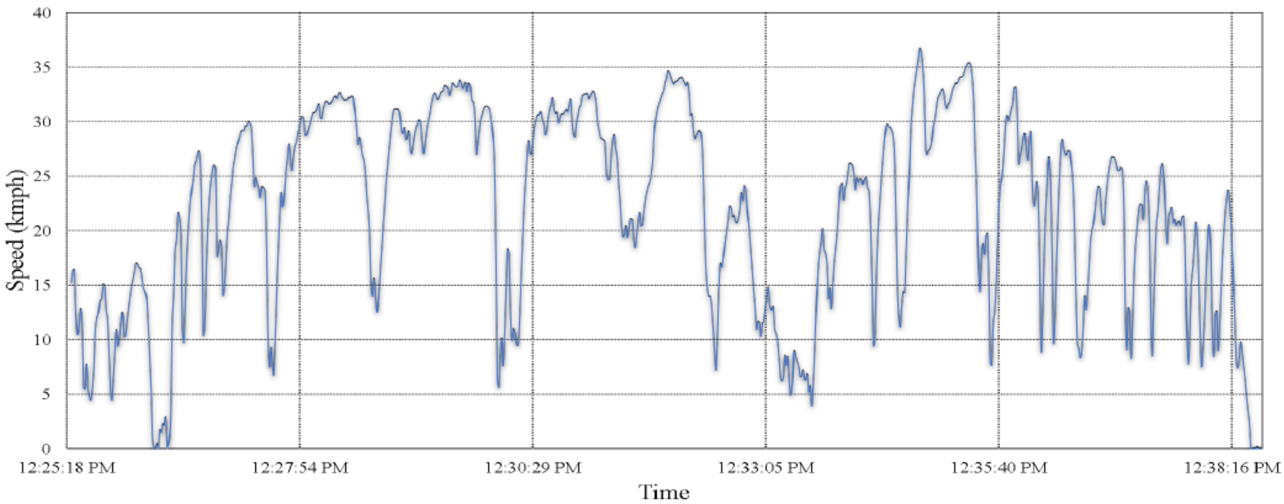

Figure 11: Speed vs Time Comparison in City Route

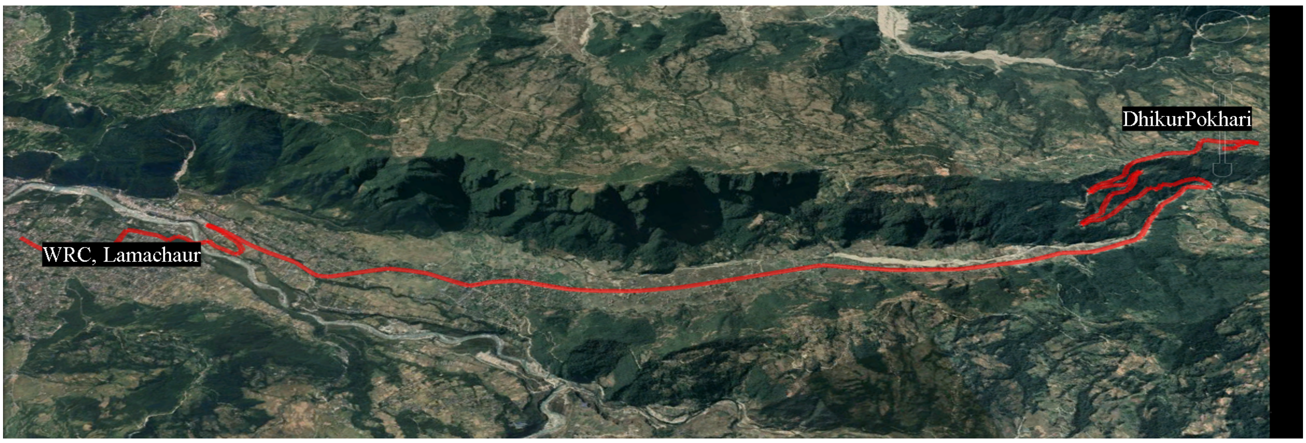

Figure 12: Visualization of Highway Route in Google Earth

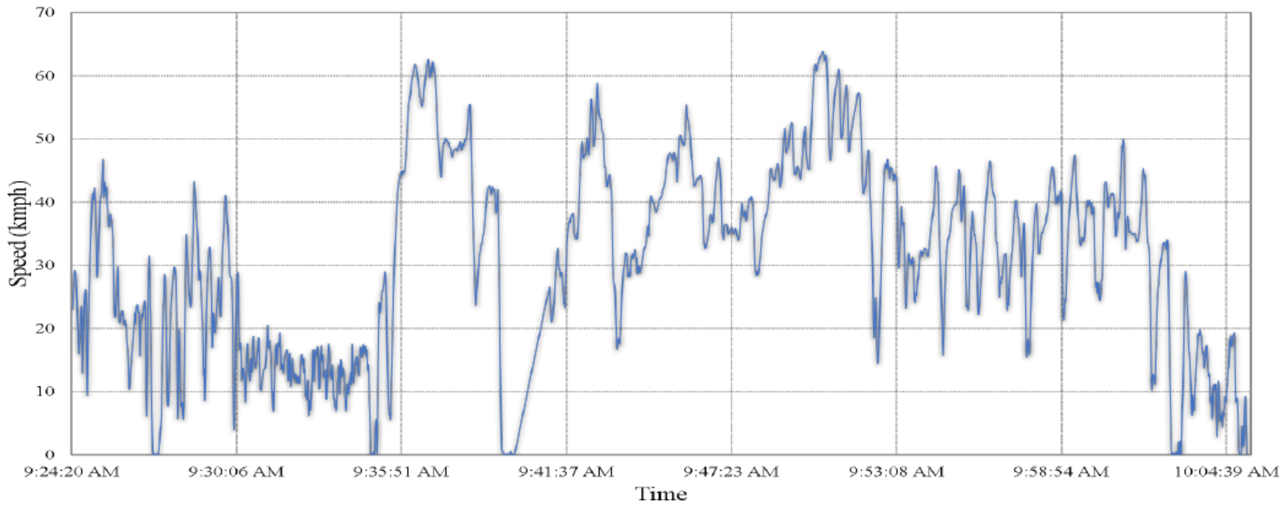

Figure 13: Speed vs Time Comparison in Highway Route

5.1 City Commute

To gather data in dynamic mode inside city limits, the GNSS data logger was mounted to a motorcycle. After completion of logging session, data was imported to the Local computer and data was examined. During the course of the logging session,the position, speed and time information was obtained from the Data Logger of the given route. The raw data in .UBX format was also recorded and obtained. First of all, the co-ordinates of points recorded along the way were connected using QGIS. Then the obtained line feature of the route was imported to Google Earth application and it was visualized as in Figure 11. A graph was used to illustrate these data where the vehicle's speed was plotted in the y-direction, while time elapsed was plotted in the x-direction as shown in Figure 12.

From the figure, the variation of speed is seen from the comparison with time as well as the distance covered.In order to examine the speed of the vehicle in relation to the condition of the traffic and road at that time, a route progression video has also been made. On further analysis, it was found that for the city commute, the decline in speed was seen mainly on the intersections depicting increase in traffic volume and a necessity to slow down or decelerate.

5.2 Highway Travel

The route followed for GNSS Data Logging in highway was from Pashchimanchal Campus, Lamachaur to Dhikur Pokhari. The speed information of the vehicle on which the GNSS Data Logger was on-board was also extracted in different instances of time in the .CSV file. The raw data in .UBX format was also recorded and obtained.The co-ordinates of points recorded along the way were connected using QGIS. Then the obtained line feature of the route was imported to Google Earth application and it was visualized as in Figure 13.A graph was used to illustrate the logged data. The vehicle's speed was plotted in the y-direction, while time was plotted in the x-direction as shown in Figure 14.

This allowed for a comparison of the vehicle's changing speed and the passing of time.The vehicle's changing speed and the progress of time and distance were contrasted.In order to examine the speed of the vehicle in relation to the condition of the traffic and road at that time, a route progression video was also made.For the highway route, it was analyzed that, the decline in speed was seen in the road sections where the road conditions are bad or off-road sections.

6. Conclusion and Recommendation

Low-cost hardware and freely accessible software were used to create a Low-cost GNSS Data Logger. It is inexpensive, portable, and dynamic in its manner of use. The acquired GNSS Data logger will be helpful for capturing raw GNSS data as well as acquiring PVT data. The Data Logger can be used in both static and dynamic application modes. For this study, data were collected in dynamic mode on the Pokhara-Baglung highway and inside city limits. Further analysis was conducted using the PVT data that were recorded, and raw data may be used for post-processing and additional analysis.The data logger was found to be particularly efficient in gathering data in a dynamic mode and performing relevant analysis.

A versatile receiver module, the ublox NEO-M8T can be used in a number of applications. In this data logger, the data was recorded using a standalone receiver module. However, a base station could be utilized to give real-time correction and obtain more accurate data than the stand-alone receiver with the usage of NTRIP. The raw GNSS data from the NEO-M8T plus data from a CORS station or other more precise receiver can be post-processed later to obtain more precise data.Besides the application of the GNSS data logger for positioning, the raw data can be used for variety of applications such as space weather,urban heat island, TEC and so on.Additionally, for the analysis portion, the data logger can be installed in a regular fleet of vehicles so that additional research and analyses linked to traffic monitoring can be carried out.

Acknowledgement

The authors are thankful to Er. Saurav Gautam and Er. Digvijaya Paudel for their support during this research.

References

[1] ICAO, ICAO, Annex 10 to the Convention of International Civil Aviation, no. October. 2001.

[2] C. J. Hegarty and E. Chatre, "Evolution of the Global Navigation SatelliteSystem (GNSS)," Proc. IEEE, vol. 96, no. 12, pp. 1902–1917, 2009, doi: 10.1109/jproc.2008.2006090.

[3] C. Cai and Y. Gao, "Modeling and assessment of combined GPS/GLONASS precise point positioning," GPS Solut., vol. 17, no. 2, pp. 223–236, 2013, doi: 10.1007/s10291-012-0273-9.

[4] M. Ge, H. Zhang, X. Jia, S. Song, and J. Wickert, "What is achievable with the current compass constellation?," GPS World, vol. 23, no. 11, pp. 29–33, 2012.

[5] X. Li, X. Zhang, X. Ren, M. Fritsche, J. Wickert, and H. Schuh, "Precise positioning with current multi-constellation Global Navigation Satellite Systems: GPS, GLONASS, Galileo and BeiDou," Sci. Rep., vol. 5, p. 8328, 2015, doi: 10.1038/srep08328.

[6] u-blox, "u-blox M8 Receiver Description Including Protocol Specificaton," 2016.

[7] T. Liu and B. Li, "Single-frequency BDS / GPS RTK With Low-cost U- blox Receivers," pp. 232–238, 2017.

[8] R. A. Oliveira, E. Khoramshahi, J. Suomalainen, T. Hakala, N. Viljanen, and E. Honkavaara, "Real-time and post-processed georeferencing for hyperpspectral drone remote sensing," Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch., vol. 42, no. 2, pp. 789–795, 2018, doi: 10.5194/isprs-archives-XLII-2-789-2018.

[9] F. Yan, X. Hu, L. Xu, and Y. Wu, "Construction and accuracy analysis of a bds/gps-integrated positioning algorithm for forests," Croat. J. For. Eng., vol. 42, no. 2, pp. 321–335, 2021, doi: 10.5552/crojfe.2021.1105.

[10] S. Rover and A. Vitti, "GNSS-R with low-cost receivers for retrieval of antenna height from snow surfaces using single-frequency observations," Sensors (Switzerland), vol. 19, no. 24, 2019, doi: 10.3390/s19245536.

[11] V. DI Pietra, N. Grasso, M. Piras, and P. Dabove, "Characterization of a mobile mapping system for seamless navigation," Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch., vol. 43, no. B1, pp. 227–234, 2020, doi: 10.5194/isprs-archives-XLIII-B1-2020-227-2020.

[12] A. Tayebi, J. Gómez, M. Fernández, F. S. de Adana, and O. Gutiérrez, "Low-cost experimental application of real-time kinematic positioning for increasing the benefits in cereal crops," Int. J. Agric. Biol. Eng., vol. 14, no. 3, pp. 194–199, 2021, doi: 10.25165/j.ijabe.20211403.5812.

[13] E. Andrikopoulou, I. Spyropoulou, H. Perakis, and ..., "Extracting Pedestrian Behavior Using Low Cost GNSS Receiver Data," Proc. 8th, 2020, [Online]. Available: https://www.clge.eu/wp-content/uploads/2019/04/GIS_Eleni_Andrikopoulou_GR_Pedestrians.pdf.

[14] L. A. Barbosa, D. C. Costa, and H. C. de Oliveira, "Evaluation of low-cost GNSS receivers for speed monitoring," Case Stud. Transp. Policy, vol. 10, no. 1, pp. 239–247, 2022, doi: 10.1016/j.cstp.2021.12.005.

[15] M. J. Wouters and E. L. Marais, "Using Low-cost Receivers for Multi-GNSS Time Transfer," IFCS/EFTF 2019 - Jt. Conf. IEEE Int. Freq. Control Symp. Eur. Freq. Time Forum, Proc., pp. 1–5, 2019, doi: 10.1109/FCS.2019.8856054.