

## COMPARATIVE STUDY BETWEEN GLOBAL POSITIONING SYSTEMS USED ON REMOTELY PILOTED AIRCRAFT SYSTEMS

**Attila Laszlo BOER, Marius Cristian LUCULESCU, Luciana CRISTEA, Sorin Constantin ZAMFIRA, Ion BARBU**

Transilvania University of Braşov, Romania (aboer@unitbv.ro, lucmar@unitbv.ro, lcristea@unitbv.ro, zamfira@unitbv.ro, [ibarbu@unitbv.ro](mailto:ibarbu@unitbv.ro))

DOI: 10.19062/2247-3173.2016.18.1.16

***Abstract:** The paper presents a comparative study on Global Positioning Systems that can be used on Remotely Piloted Aircraft Systems (RPAS). Neo-M8N and Neo-6 from U-blox were taken into account. The study focuses on the interfacing with a microcontroller development system and errors positioning data obtained under the same conditions. Experimental data show that receiving conditions have significant impact on the accuracy of results. For position and altitude the NEO-M8N receiver provides higher accuracy under good receiving conditions. The compass calibration is mandatory before bearing measurements.*

***Keywords:** Remotely Piloted Aircraft System, Global Positioning System, microcontroller.*

### 1. INTRODUCTION

The Global Positioning System (GPS) is a satellite-based navigation system that was developed as a military application in the 1970s. GPS is a one-way-ranging system, i.e. users can only receive the signals [1].

GPS consists of a constellation of satellites, known as Initial Operational Capability. The satellite orbits are nearly circular and their period is about 12 hours.

GPS involves three major components:

- the space segment, which consists of the 24-sattelite constellation;
- the control segment, which consists of a worldwide network of tracking stations;
- the user segment, which includes all users.

Using a receiver the user can determine his position anywhere in the world (providing that GPS signal is available). The signal contains the coordinates of the satellites as function of time. GPS ensures very accurate timing and frequency signals using atomic clocks on the satellites [1-3].

The precision of GPS data depends on both the GPS receiver and the observing conditions. In the last years several studies were published regarding GPS data accuracy and reliability [4-7]. In the present work we will compare the accuracy of GPS data obtained with the NEO-M8N and NEO-6M GPS receivers, which can be used on Remotely Piloted Aircraft Systems.

### 2. THE NMEA PROTOCOL

NMEA stands for the National Marine Electronics Association. In 1983 the association adopted the NMEA 0183 format for interfacing marine electronic devices. It

has been updated several times, the latest release being v4.10 published in early May 2012.

The NMEA data streams include information on latitude, longitude, time, altitude and other variables. The data is sent in form of sentences. Each sentence starts with a dollar sign "\$" and ends with a carriage-return line-feed <CR><LF>.

In the present article we will restrict our discussion to one sentence only, the GGA: Global Positioning System fix data. The general structure of the GGA sentence has the following form [1,2]:

```
$GPGGA,092750.000,5321.6802,N,00630.3372,W,1,8,1.03,61.7,M,55.2,M,,*76
```

We are interested in fields 3+4 (latitude), 5+6 (longitude) and 10+11 (altitude). The advantage of the NMEA standard is that being an ASCII format it is very easy to parse using standard command line tools such as awk, perl or python (for more details see section 3).

Most GPS receivers available on the market support the NMEA 0183 standard.

### 3. HARDWARE AND SOFTWARE SETUP

Our hardware setup consists of a standard GPS receiver connected to a Raspberry Pi 2 board via serial and i2c interfaces.

The Raspberry Pi is a single-board computer developed by the Raspberry Pi foundation. During the last years several generations have been released. The second generation, which has been used in our hardware setup, was released in February 2015. All models feature an ARM compatible CPU. The board uses SD cards to store the operating system (mainly Linux variants) and programs. Lower level input-output is provided by the GPIO pins (general purpose input/output) along the edge of the board. The pins support common protocols such as serial or i2c. In our setup the serial interface was used for the GPS data and the i2c interface for the compass [8,9].

As GPS receivers the NEO-6M and NEO-M8N have been used. The NEO-6 series has been designed with low power consumption and low cost in mind. The NEO-M8N GPS receiver incorporates the HMC5883L digital compass and includes an internal Flash that allows firmware updates. Both modules work with a baud rate of 38400 and incorporate power and fix indicator LEDs [10,11].

In order to display the NMEA protocol sentences transmitted by the GPS receiver via the serial port (/dev/ttyAMA0 in the case of the Raspberry Pi board) we can use the following bash script:

```
#!/bin/bash
i="0"
while [ $i -lt 200 ]
do
    OUTPUT=$(cat /dev/ttyAMA0 | ./script1.sh | ./script2.sh)
    if [[ ${OUTPUT} ]]; then
        echo ${OUTPUT}
        i=$((i+1))
    fi
    sleep 1
done
```

script2.sh uses the standard UNIX command line utility "awk" to parse the NMEA sentences and to extract the information regarding latitude, longitude and altitude.

```
awk -F, '\$GPGGA/ {print strftime("%Y-%m-%d "), $2, (substr($3,0,2) +
(substr($3,3) / 60.0)), $4, (substr($5,0,3) + (substr($5,4) / 60.0)), $6, $10; fflush();}'
```

The digital compass is connected via the i2c interface at address 0x1e. Data acquisition has been done using the following Python script [12,13]. It outputs the coordinates x, y and z.

```
#!/usr/bin/python
import smbus
import time
import math
import sys

bus = smbus.SMBus(1)
address = 0x1e

def read_byte(adr):
    return bus.read_byte_data(address, adr)

def read_word(adr):
    high = bus.read_byte_data(address, adr)
    low = bus.read_byte_data(address, adr+1)
    val = (high << 8) + low
    return val

def read_word_2c(adr):
    val = read_word(adr)
    if (val >= 0x8000):
        return -((65535 - val) + 1)
    else:
        return val

def write_byte(adr, value):
    bus.write_byte_data(address, adr, value)

write_byte(0, 0b01110000)
write_byte(1, 0b00100000)
write_byte(2, 0b00000000)

for i in range(0,500):
    x_out = read_word_2c(3)
    y_out = read_word_2c(7)
    z_out = read_word_2c(5)
    print x_out, y_out, z_out
    sys.stdout.flush()
    time.sleep(0.1)
```

Bearing in the x-y plane can be computed as:

$$\alpha = \text{atan}\left(\frac{y}{x}\right) \quad (1)$$

This value should be corrected taking into account the so called magnetic declination, i.e. the angle on the horizontal plane between magnetic north and true north.

#### 4. EXPERIMENTAL RESULTS

We studied the accuracy of GPS data provided by the NEO-6M and NEO-M8N receivers. A number of approximately 200 measurements have been taken both in good and difficult receiving conditions.

Table 1 shows the standard deviation of latitude, longitude, altitude and bearing, respectively the localization error in meters. As we can observe receiving conditions and radio interference might have a very pronounced impact on the accuracy of results.

The localization error was computed using the following formula:

$$d = R\sqrt{a^2 + b^2} \tag{2}$$

where  $R$  is Earth's radius (6371 km) and

$$a = \Delta\lambda \cdot \cos\varphi_m \tag{3}$$

$$b = \Delta\varphi \tag{4}$$

Where  $\varphi$  being the latitude and  $\lambda$  the longitude and  $\varphi_m$  denotes the mean value of latitude.

Table 1. Standard deviation of latitude, longitude, altitude and bearing, respectively the localization error

GPS receiver	STDEV for latitude (seconds)	STDEV for longitude (seconds)	STDEV for altitude (m)	Localization error (m)	STDEV for bearing (degrees)
NEO-6M (good conditions)	< 0.18	< 0.18	10.44	< 6.783	0.1253
NEO-6M (difficult conditions)	0.688	1.547	8.29	39.597	0.2172
NEO-M8N (good conditions)	< 0.18	< 0.18	4.26	< 6.783	0.4762
NEO-M8N (difficult conditions)	0.218	0.784	19.78	18.229	1.1965

Figure 1 shows the values for the altitude during the 200 measurements for both the NEO-6M (red) and NEO-M8N (blue) modules. We must note that air pressure sensor is the most accurate way to measure altitude, however it needs to be calibrated whenever a reliable reference is available or when at sea level.

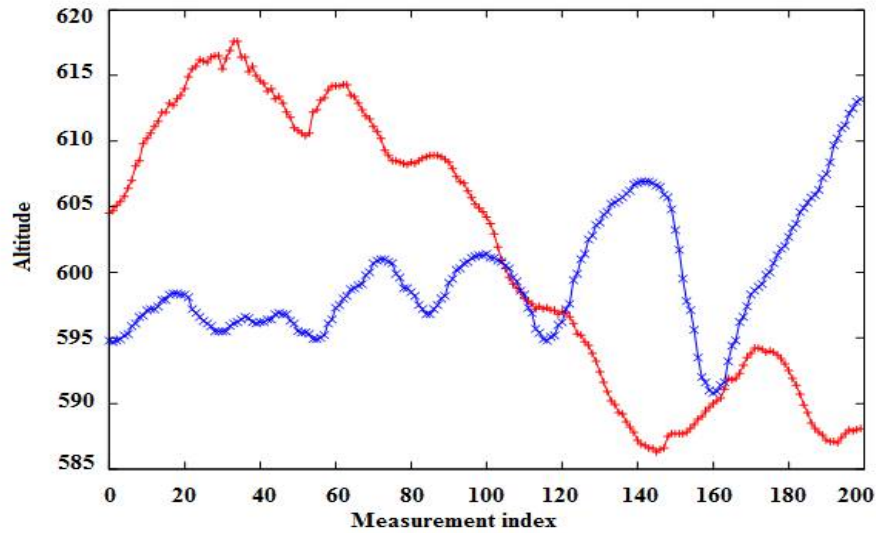


FIG. 1. Values for the altitude for both the NEO-6M (red) and NEO-M8N (blue) modules.

In order to display accurate values for the bearing, we must be calibrate the compass. We start by rotating the device in the horizontal plane and we register the values for the  $x$  and  $y$  coordinates. If we plot the values we notice that the circle isn't centred around the origin (see Fig. 2).

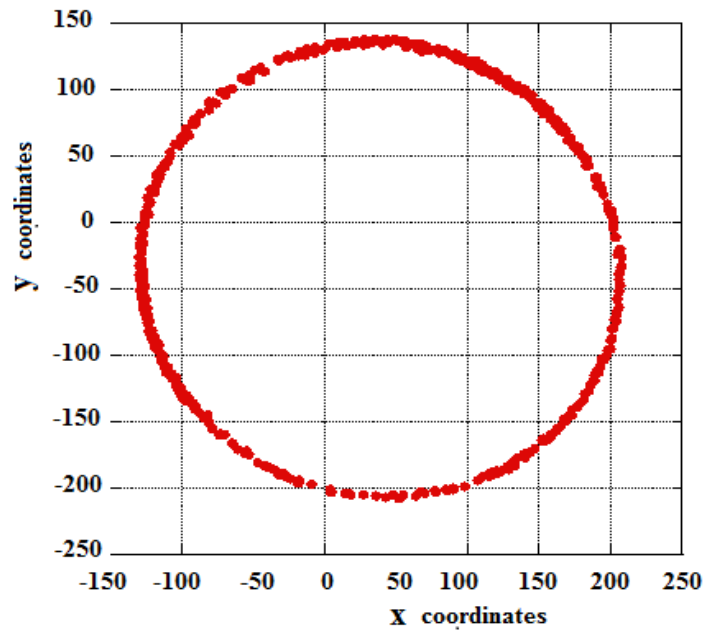


FIG. 2. x and y coordinates in compass calibration

The  $x$  and  $y$  offset can be computed as:

$$x_{offset} = \frac{x_{min} + x_{max}}{2} \quad y_{offset} = \frac{y_{min} + y_{max}}{2} \quad (5)$$

Table 2 shows the offset values for both GPS receivers. We note that the NEO-6M module has an unusually high offset along the  $y$  axis.

Table 2. The offset values for both GPS receivers

GPS receiver	$x_{offset}$	$y_{offset}$
NEO-6M	38	-170
NEO-M8N	39	-35

## 5. FINAL REMARKS

We studied GPS data accuracy for the NEO-6M and NEO-M8N receivers. The hardware setup consists of a GPS receiver, which connects to a Raspberry Pi 2 board via serial and i2c interfaces. Experimental data shows that receiving conditions have significant impact on the accuracy of results.

For position and altitude the NEO-M8N receiver provides higher accuracy under good receiving conditions. We have shown that compass calibration is mandatory before bearing measurements.

For the NEO-M8N the offsets along the x and y axis were moderate, however for the NEO-6M unusually high y offset was detected.

## ACKNOWLEDGMENT

This paper was realized within the Partnerships Programme in priority domains-PN-II, which runs with the financial support of MEN-UEFISCDI, Project no. PN-II-PT-PCCA-2013-4-1629.

## REFERENCES

- [1] Ahmed El-Rabbany, *Introduction to GPS – the global positioning system*, Artech House, Boston, London (2002).
- [2] E. D. Kaplan, C. J. Hegarty, *Understanding GPS - Principles and Applications*, Artech House, Boston-London (2006).
- [3] G. Xu - *GPS - Theory, Algorithms and Applications* 2nd ed., Springer Verlag (2007).
- [4] C. Cai et. al., *An analysis on combined GPS/COMPASS data quality and its effect on single point positioning accuracy under different observing conditions*, *Advances in Space Research* 54 818-829 (2014).
- [5] G. I. Emel'yantsev et. al., *Improving the accuracy of GPS compass for small-sized objects*, *Gyroscopy and Navigation* 6 166-171 (2015).
- [6] X. Li et. al., *Accuracy and reliability of multi-GNSS real-time precise positioning: GPS, GLONASS, BeiDou, and Galileo*, *Journal of Geodesy* 89 607-635 (2015).
- [7] K. Liu et. al., *Improving Positioning Accuracy Using GPS Pseudorange Measurements for Cooperative Vehicular Localization*, *IEEE Transactions on Vehicula* 63 (2014).
- [8] P. Hongyan et. al., *Drive design for ship GPS navigation equipment based on Linux operating system*, International Conference on Educational and Network Technology (ICENT) (2010).
- [9] S. Monk, *Programming the Raspberry Pi - Getting started with Python*, Mc Graw Hill (2013).
- [10] NEO-6 u-blox 6 GPS Modules Data Sheet. Available at [https://www.u-blox.com/sites/default/files/products/documents/NEO-6\\_DataSheet\\_\(GPS.G6-HW-9005\).pdf](https://www.u-blox.com/sites/default/files/products/documents/NEO-6_DataSheet_(GPS.G6-HW-9005).pdf), accessed on 02 April 2016
- [11] NEO-M8N u-blox GNSS module Data Sheet. Available at [https://www.u-blox.com/sites/default/files/NEO-M8N-FW3\\_DataSheet\\_\(UBX-15031086\).pdf](https://www.u-blox.com/sites/default/files/NEO-M8N-FW3_DataSheet_(UBX-15031086).pdf), accessed on 02 April 2016
- [12] Mark Lutz, *Learning Python*, Publisher O'Reilly, (2013).
- [13] Connecting and calibrating a HMC5883L Compass on the Raspberry Pi, <http://blog.bitify.co.uk/2013/11/connecting-and-calibrating-hmc5883l.html>, accessed on 02 April 2016