A GPS Receiver Designed for Cubesat Operations

Eamonn P Glennon
UNSW Sydney

Kevin Parkinson
UNSW Sydney

Nagaraj Cs
UNSW Sydney

Yong Li
UNSW Sydney

A GPS Receiver Designed for Cubesat Operations

Éamonn Glennon [1], Kevin Parkinson [2], Peter Mumford [1], Nagaraj Shivaramaiah [1]
Yong Li [1], Rui Li [1] and Yuanyuan Jiao [1]

[1] *School of Surveying and Spatial Information Systems (SSIS)*
University of New South Wales, Sydney, NSW 2052, Australia
[2] *General Dynamics Corporation Limited, Albany, New Zealand*

Summary: The Namuru V3-2 is an L1 FPGA based GNSS receiver designed for integration into a Colony II cubesat. This paper describes the receiver design requirements, as well as the hardware and software design of the subsequent implementation. Preliminary performance results obtained using the receiver are also presented.

Keywords: Space qualified GPS receiver, Colony 2 Cubesat, Namuru V3.2

Introduction

Australia has recently developed a renewed interest in space related activities with the establishment of the Australian Space Policy Unit. As part of this interest, a research program has been initiated via the Australian Space Research Program (ASRP) funding grants. The School of Surveying and Spatial Information Systems (SSIS) at the University of New South Wales (UNSW) was successful in gaining ASRP funding for the "Synthetic Aperture Radar (SAR) Formation Flying" project, also known as the Garada. As the name suggests, the program involves undertaking research into flying satellites in formation. The project has several industry partners and collaborators, including Defence Science and Technology Organisation (DSTO), BAE Systems, Astrium, and General Dynamics Corporation Ltd (New Zealand). The Garada collaboration has led to SSIS involvement in another collaborative cubesat mission spearheaded by DSTO, known as Biarri. The Namuru V3.2 GPS receiver described in this paper is aimed at satisfying the requirements for Biarri and Garada.

Operating a Global Positioning System (GPS) receiver in low earth orbit (LEO) represents a significant challenge compared to normal terrestrial use. One problem is that commercial off the shelf (COTS) receivers are optimized for terrestrial operation. This means that software algorithms embedded in the devices are not tuned to accommodate the large variations in the received signal Doppler frequencies that are experienced in low earth orbit (LEO) or the rapid changes in satellite visibility that accompany the orbital motion. Even if the receiver is able to acquire and track the GPS signals, International Traffic on Arms Regulations (ITAR) requirements often result in the receiver not producing navigation outputs because the orbit altitude and speed exceed the ITAR limits of 60,000 feet and 1,000 knots. The harsh environment of space with its vacuum, radiation and temperature cycling caused by continual entry and exit from the sun's heat causes difficulties for a standard design, as does the extreme vibration that the receiver is subject to during launch. Care must also be taken in the thermal design of the equipment in to avoid overheating the components, as well as the incorporation of low thermal resistance pathways to allow any excessive heat to be conducted away. This follows because heat can only be removed by radiation in a vacuum and must be conducted to parts of the spacecraft where this may occur. There are also power-budget constraints as well as specific mission requirements that may not be satisfied by a COTS receiver.

All of these factors ensure that a full custom design is worthy of consideration, despite the challenges imposed by such an undertaking. This paper describes the requirements imposed on the Namuru V3.2 GPS receiver, discusses the receiver itself and also provides some estimates of the performance under space conditions based on results obtained using the Namuru V2R4 receiver design.

Biarri Program

The GPS/GNSS subsystem represents a vital component in both the Biarri and Garada programs, although in this paper it is generally to Biarri we will refer. This is because Garada is a university driven program that may be more properly considered as a feasibility study, with a strong research element. The outcomes of this study are yet to be determined and the requirements are therefore not fully defined, although some of the requirements for components such as GPS can be estimated based on similar projects by other research groups.

In comparison to Garada, Biarri is an international defence-science collaborative program, with the US, Canada, the UK, and Australia each contributing a subsystem of the mission. One of the benefits of such collaboration is the capability building process that it fosters in the participating nations, especially as it relates to space related programs. This program involves integrating several payloads from the participating nations into three Colony 2 3U-cubesats that will be supplied by the USA [1], with the system integration of the payloads being carried out by AFRL. This mission is one of several for which the Colony 2 Bus (C2B) will be employed, with another example being the STARE cubesat mission [2].

The Australian contribution to this collaboration was selected by DSTO, taking into account the available capabilities and the requirements of the mission. There are two Australian contributions, namely the development and supply of a GPS receiver through SSIS and the supply of a satellite laser ranging (SLR) capability through Electro Optic Systems (EOS) Pty Ltd. This selection of subsystems reflects one of the technical goals of the mission, which is to demonstrate an ability to determine the relative separation between multiple cubesats with an accuracy of approximately 10 cm. The availability of SLR functionality to each of the three spacecraft comprising the formation will also permit valuable validation between the two systems to be undertaken. In addition, the GPS is also required to provide precise timing functionality at the 20 ns level to other subsystems in the spacecraft.

Namuru GPS Hardware

Namuru V1 and V2

SSIS has been active in the field of field programmable gate array (FGPA) GPS receivers since 2006 when the Namuru V1 GPS receiver was introduced [3, 4], although development of the receivers started in late 2004 [5]. This first generation device consisted of an Altera Cyclone II FPGA, a Zarlink GP2015 radio frequency (RF) front end [6], SRAM and Flash memories, EEPROM and a real-time clock. A NIOS-2 soft processor, and 12 GPS tracking channels with an interface similar to that employed by the Zarlink GP2021 baseband processor [7] were implemented using the FPGA hardware fabric to perform baseband processing of the GPS IF signals. The firmware used to perform the said processing was based on the GPS Architect codebase, also from Zarlink Semiconductor.

Although the Namuru V1 receiver was well received, there were a number of applications for which it was ill suited, with one of these being any application requiring two separate RF front ends. This limitation was remedied with the subsequent introduction of the Namuru V2 receiver, which is still in limited production even though new receivers are nearing design completion. The Namuru V2 differed from the V1 with the inclusion of a second GP2015 RF front end, an L2 to L1 mixer component, replacement of SRAM with a much larger SDRAM, an upgrade to a larger Cyclone II, inclusion of optional MEMS 3-axis gyroscope and accelerometers and the replacement of wired Ethernet with a USB2 connection. The receiver has found widespread use for GPS research, including research into cross correlation mitigation techniques [8], reception of new GNSS signals and has even found use for research into space-based navigation by DLR in Germany [9]. This space related research has included the generation of high precision carrier phase observations, the use of the receiver for GNSS reflectometry and the use of the receiver on board a sounding rocket.

Namuru V3.2 Hardware

From a hardware perspective, there were several reasons why the Namuru V2 receiver was unsuitable for direct use on a cubesat platform, notwithstanding the excellent work by DLR [9]. Some of these are itemized below.

1. The 170 × 99 mm dimensions of Namuru V2 were too large for the Colony 2 cubesat.
2. The Namuru V2 power consumption exceeded cubesat desired power limit of 1 W by a significant margin.
3. Some components included on the Namuru V2, such as the battery that were used to power the real time clock (RTC) and battery backed memory, were not appropriate for an extended mission in space.
4. Some components included on the Namuru V2, such as one of the 2 GP2015 front ends and the USB2 functionality, were unnecessary.
5. The Namuru V2 did not have the cubesat required communication interface (RS422).
6. The Cyclone II FPGA would be susceptible to single event upsets (SEU) caused by exposure to radiation and corruption of the FPGA configuration stored within the FPGA RAM.
7. The Namuru V2 design does not allow for latch up detection and mitigation.

Because of these difficulties, a new Namuru V3.2 GPS receiver hardware design has been in development by Parkinson of General Dynamics Corporation Ltd (New Zealand). SSIS recently took delivery of several new receiver prototypes, as shown in Fig. 1, with the firmware and FPGA designs currently being ported across to the new platform. This development has followed the design philosophy of using commercially available components rather than fully space- qualified components in order to constrain the cost of the device. The DLR Phoenix-XNS GPS receiver, which is a modified SigNav MG5001 GPS receiver manufactured in Australia [10], is an excellent example of the success of this approach having been used in several successful DLR space missions [11].

Some of the features of the Namuru V3.2 are [12]:

1. The mechanical dimensions of 82 × 82 mm are tailored to the Colony 2 bus (C2B).
2. The electrical interface includes the RS422 connections required by the C2B, as well as the particular I/O signals required for the mission.
3. The design employs a capable processor in the form of an Actel SmartFusion A2F500 system on a chip that includes a hard wired ARM Cortex M3 processor, 64 kB of internal SRAM, 512 kB of internal Flash, a Flash based FPGA fabric with 11,520 FPGA tiles, and other peripherals.

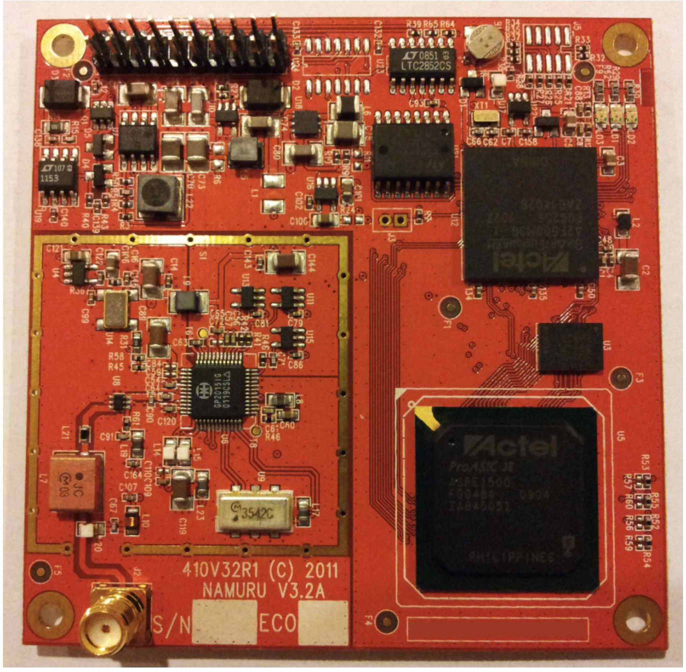

Fig. 1: Loaded Namuru V3.2A printed circuit board

4. The design includes an Actel ProASIC3E Flash based FPGA with 38,400 tiles for the inclusion of large digital hardware designs.
5. A serial Flash memory has been included for non-volatile storage and the storage of FPGA and firmware images required for over-the-air reprogramming.
6. A Zarlink GP2015 L1 RF front end has been used because it is a well-understood front end, even though the device is no longer in production. The RF portion of the receiver includes an RF shield to reduce electromagnetic susceptibility.
7. A 10 MHz voltage controlled temperature compensated crystal oscillator (VC-TCXO) has been used in order to discipline the 10 MHz local clock and eliminate saw-tooth timing errors [13].
8. 1 MB of fast external SRAM has been included in order to ensure that sufficient SRAM is available for firmware and data variables.
9. The design includes latch up detection on the power supply that causes the power to be cycled if the current drain exceeds a set threshold.
10. Mechanical stresses caused by thermal cycling have been considered in the design of the printed circuit board through the use of solid (filled) vias and the use of transitional tin-lead solder alloys.
11. A super-cap allows the RTC included in the SmartFusion A2F500 to be run for at least 24 hours.
12. An SMA antenna connection has been employed to allow the antenna to be robustly connected to the receiver.

When the SmartFusion A2F500 FPGA processor was selected, it was hoped that the FPGA fabric present in the device would be sufficient to include at least 12 tracking channels. Unfortunately, it was subsequently discovered that the current baseband implementation does not fit within the capacity of the largest device in the family and it was decided to include an additional FPGA with 3 times the capacity of the SmartFusion FPGA. It is hoped that the creation of a future optimized correlator design that employs time multiplexing techniques may allow a full GPS correlator to be included entirely within the A2F500, thereby eliminating the need for the additional part. Actel Flash based FPGAs were selected because once the design is programmed into the devices, those designs are expected to show improved immunity to the single event upsets (SEUs) and latch-ups (SEL) that can occur in a radiation affected environment [14].

Namuru Firmware

One of the past deficiencies of the Namuru based receivers has been the lack of in-house developed firmware. Instead, the firmware that was used for the receiver was a port of the GEC Plessey GPS Architect code [15, 16] designed for the Zarlink GP2015 and GP2021 GPS chipsets [6, 7]. One reason for this choice was that at the time, there was no other GPS firmware code base that was available as full source code and targeted to a chipset with published datasheets. The problem of lack of access to GPS datasheets continues to this day, with many GPS manufacturers refusing to provide access to the custom chipsets that they employ. This makes it virtually impossible for a third party to write new firmware for these platforms. A second reason is that the amount of software engineering required to produce a reliable firmware suite able to perform the many functions of a GPS receiver is substantial. It also requires specialized knowledge of GPS that few firmware engineers hold.

Unfortunately, although the GPS Architect source code represented an excellent starting point for the Namuru firmware, other difficulties accompany it. First, the source code was supplied as part of a development kit that is no longer available from Zarlink, and who have exited the GPS market. This represents a significant problem because Zarlink owns the Architect code and it was never placed into the public domain, making further distribution of code derived from the GPS Architect source code problematic. Second, the main purpose of the GPS Architect was to create a simple reference design that could be used to demonstrate the functionality of the chipset and that could be used as a starting point for Zarlink customers. As a result, the functionality of the code was somewhat limited, as was its performance. Third, the source code was highly coupled to the chipset to which it was targeted, making it difficult to incorporate new front ends or changes to the correlator design.

In order to remedy these problems, Glennon started working on a new suite of firmware targeted at the Namuru V2 whilst undertaking his part time PhD research at SSIS [17] and continued to work on this firmware following his recruitment by the university as a researcher. As a result, firmware different to the GPS Architect is now available to the SSIS for the Namuru platform. Important features of the firmware, called Aquarius, are listed below.

1. The firmware is written in ANSI C and is currently targeted at the Altera NIOS-2 processor.
2. The code is reasonably well commented.
3. A simple real time operating system (RTOS) has been written to allow the firmware to be split into different priority threads, to provide signals between different tasks and semaphores for the protection of shared memory structures.
4. A GPS driver layer has been written to hide some of the implementation related features of the RF front end and baseband correlator, thereby limiting the impact of changes to these components to the driver layer. In theory, it should be possible to change only the GPS driver in response to a hardware design change in which a new RF front end or new correlator is introduced.
5. Closely related firmware modules (*.c* and *.h* files) have been grouped into a directory hierarchy that should assist the programmer in locating modules of interest. The directories include an *Rtos* directory for the RTOS, a *GpsTrk* directory for GPS driver, tracking and measurement functionality, *NavData* for extraction of navigation data and the decoding of navigation messages, *Timing* for timing related functions and *SvDB* for functions related to satellite selection, the storage and retrieval of satellite

Table 1: Firmware task structure

Task	**Priority**	**Remark**
TestTask1	240	*Test task – optional task*
PreDSPTask	195	Performs pre-integration & sample counting
TrackerTask	190	Tracking, reacquisition & acquisition
RtcTask	187	Update & maintain RTC
MsmtTask	185	Make observations & upload measurements
BufferBitsTask	175	Convert I&Q samples to bits and buffer
ExtractTask	170	Extract navigation message & decode data
ReportTask	150	Transmit serial port reports
CmdTask	150	Process input serial port commands
AutoTest	120	Perform automatic TTFF & reacquisition tests
TestTask3	100	*Test task – optional task*
PVTFixTask	15	Perform navigation solutions
SvDbTask	10	Calculate SV reference position, velocity & acceleration
SvSlctTask	10	Select satellites and allocate to hardware channels
ISR		**Remark**
CorrelatorISR	n/a	Measurement TIC and sample poll ACCUM_INT
TimerISR	n/a	1 ms RTOS timer

almanac, ephemeris, and user related data. A *PVTNav* directory contains code related to the calculation of position, velocity and timing (PVT) navigation solutions, while serial port message processing related code is stored in the *UserIF* directory.

6. The processing is performed in a multi-threading environment, with more frequent time critical processing performed at a higher priority than less frequent tasks. Table 1 provides some details as to the various tasks that are run in the receiver.
7. Serial port command and report messaging is performed using sentences inspired by the National Marine Electronics Association (NMEA) standard 0183.
8. The firmware is able to perform carrier phase smoothing on the pseudoranges [18].
9. The firmware is able to output L1 carrier phase measurements at 1 Hz for performing high accuracy positioning. This feature is required for both Garada and Biarri.
10. The tracking loop has been upgraded to a 3rd order phase locked loop with 2nd order frequency assist. The use of a 3rd order PLL is necessary in order to avoid carrier phase biases that would otherwise occur because of the orbital acceleration.
11. The firmware is able to track and navigate using L1 C/A code GPS satellites, as well as the newly launched Japanese QZSS Michibiki spacecraft. The use of use of Space Based Augmentation System (SBAS) such as the Wide Area Augmentation System (WAAS) is currently not supported.

Work is currently underway to port this firmware from the Altera NIOS-2 processor to the ARM Cortex M3 that is used within the SmartFusion A2F500.

Firmware Specific Features for LEO Operation

A satellite in low earth orbit will typically circle the earth many times in a single day; a simple fact has a profound effect on a GPS receiver operating on board such a satellite. One consequence is that the satellite visibility changes rapidly, which means that the satellite selection process that occurs within the space-based receiver is required to operate at a higher frequency than it does for a terrestrial receiver. The situation is further complicated when the receiver is powered up because the calculation of satellite visibility using previously stored almanac, ephemeris and location plus an estimate of the current time obtained from an on-board RTC cannot be employed. One solution to this problem is to take a previously stored satellite position and velocity vector at a known time instant and to use an orbit propagator to estimate the receiver position at the time instant obtained using the RTC. A two-line element (TLE) description of the satellite orbit previously provided to the receiver could also be used in place of the previously saved position and velocity vectors. This approach has the disadvantage of requiring the inclusion of additional code within the GPS firmware. There may also be processing constraints caused by the use of small fixed-point processors on board the GPS receiver that limit the time interval over which such a prediction may be carried out. However, this may not matter given that the calculation need only be carried out at startup.

An alternative solution to this problem is to avoid making assumptions about the location of the receiver when it is powered up. This implies that every receiver startup is a cold-start, which requires that the receiver blindly search through all the satellites in the constellation until it has found sufficient satellites to navigate. Unfortunately this too suffers from a rather insidious problem, namely the high range of Doppler frequencies that the received carrier signal experiences that is caused by the high velocity of the LEO spacecraft. Unlike a terrestrial receiver that typically experiences a Doppler frequency variation in the range of ±4 kHz, a space-based receiver may experience a Doppler variation of ±40 kHz. This increases the search range for each blind search by a factor of 10 and causes the time to acquire each satellite to increase proportionally. Compounding this difficulty are the rapid changes to satellite visibility that may result in a previously acquired signal being lost before it has had a chance to be used to calculate a navigation solution. It is therefore vitally important to ensure that a space based GPS receiver is capable of rapidly searching a wide range of Doppler frequencies.

Several measures have been taken to resolve these difficulties in the Namuru/Aquarius receiver. The first is to trade off acquisition sensitivity against search speed by decreasing the acquisition pre-detection integration period from 4 ms to 1 ms. This increases the carrier search bandwidth of each code scan from 250 Hz (±125 Hz) to 1 kHz (±500 Hz). In addition, the number of 0.5 chip separated code phase taps included in each tracking channel has been increased from 3 to 9. These measures increase the search speed by a factor of 12, which ensures that the receiver can operate in LEO even if every startup is a cold start. In addition, work has recently been undertaken to integrate the Dundee C code implementation of the two line element (TLE) SGP4 orbit propagator into the firmware [19], which should assist in improving time-to-first-fix performance provided the receiver can be regularly supplied with updated TLE elements.

Another minor improvement that can be applied to the receiver is to take advantage of the known orbit dynamics within the Kalman filter. Because the acceleration of the receiver can be predicted quite accurately using a simple gravitational model with corrections for earth flattening effects, the prediction step of the Kalman filter can propagate the position.

Preliminary Test Results

Although the Aquarius firmware has been in part-time development since mid 2006, it is only recently that the firmware has reached a stage where it has had sufficient reliability for systematic testing to be undertaken. The results that follow supplement the informal bench testing that has been undertaken through much of the development, although unlike those tests they employ a Spirent STR6560 GPS simulator [20]. Using a GPS simulator allows the receiver position, velocity, acceleration and time to be controlled, as well as the satellite constellation and the signal levels of each of the satellites, while eliminating environmental nuisance factors such as multipath. This is vital for unusual flight dynamics, such as testing the receiver for in-orbit operation, where live testing is not feasible. The magnitude of

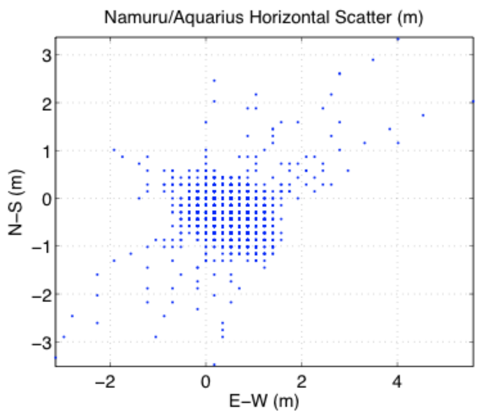

Fig. 2: Simulator static test

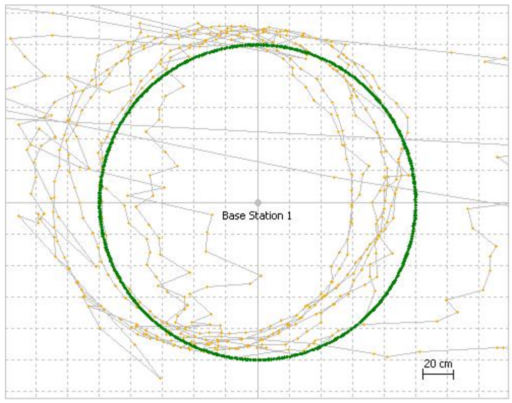

Fig. 3: Simulator carrier phase test

atmospheric errors such as ionospheric and tropospheric errors can also be fully controlled by allowing them to be defined by the very models that are used to correct for these errors in the receiver. This allows the user to verify correct operation of the models within the receiver, either by direct comparison of the model output with the truth data provided by the simulator or by observing the absence of any bias in the receiver output after application of the associated corrections.

Static Simulator Pseudorange Positioning Test

A static pseudorange positioning test is designed to test for absolute accuracy in the absence of multipath errors using strong signals. For this test, simulator ionospheric errors should be fully corrected by the standard Klobuchar ionospheric correction model provided by GPS. Similarly, the correction model of the receiver should for the most part correct the STANAG 4294 tropospheric errors generated by the simulator. For these tests, the receiver is configured for a moving scenario, which is typical for standard vehicle dynamics and a 1 second carrier smoothing interval is employed. This level of smoothing is significantly less than is often used for static scenarios, so a higher level of noise in the output solution is to be expected. The results of this test are shown in Fig. 2, where it can be seen that there is a bias of (0.46,-0.32,-1.55) and standard deviations of (0.4,0.4,1.0) meters in the east, north and up directions, with occasional outliers of up to 4 meters.

Slowly Moving Simulator Carrier-Phase Real-Time-Kinematic Test

An STR6560 GPS simulator may also be used to test carrier phase based navigation. For this test, the simulator was setup for a scenario where the receiver moves in a circle of radius 1 m at a slow speed. The simulated base station for the generation of the simulated RTCM differential corrections, which includes carrier phase information, is placed at the centre of the circle. The GPS receiver is then allowed to navigate and the raw observations, which include measured pseudoranges, Doppler frequencies, carrier phase measurements and signal strengths for each satellite, as well as received broadcast ephemeris, ionospheric corrections and UTC corrections are captured using a serial port terminal program. Similarly, the RTCM differential corrections from the base station are also captured using a serial port terminal program. As was the case for the static simulator scenario, carrier phase smoothing of the raw pseudoranges was limited to a 1 second time interval.

High precision carrier phase positioning is performed by post-processing the captured data files using the open source RTKLIB software package [21]. This is necessary because the Aquarius firmware is not able to perform such calculations internal to the receiver. To perform the required post-processing, it is first necessary to convert the output from the

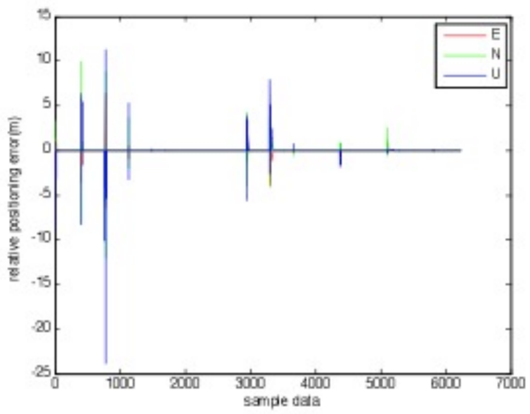

Fig. 4: Spacecraft relative positioning error

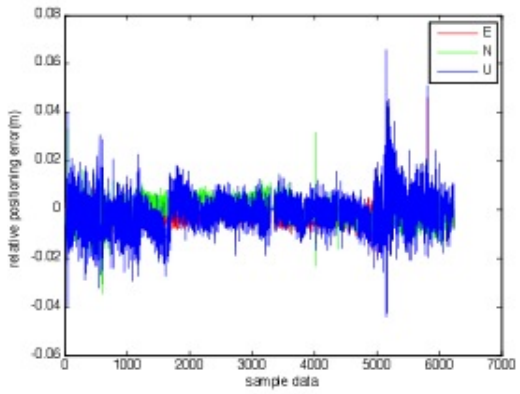

Fig. 5: Spacecraft relative positioning error with outliers removed

Namuru and the simulator into RINEX files. The Namuru output is converted to RINEX using an 'ad-hoc' utility created for the purpose, while the simulator RTCM output is converted to RINEX using the 'rtkconv.exe' utility included a part of the RTKLIB software suite. Once two sets of RINEX files have been obtained, it is possible to use the 'rtkpost.exe' utility to perform the high precision carrier phase differential positioning. Some configuration values of the 'rtkpost.exe' program parameters were adjusted before performing the RTK post-processing and the program was run in the 'kinematic' mode. Fig. 3 shows the post-processing results, where the orange dots indicate solutions obtained using a float-solution, while the 'green' dots indicate the improved precision solution once integer ambiguity resolution has successfully taken place and the integer ambiguities have been fixed. The high precision obtained from the carrier phase solution is clearly apparent and demonstrates that the receiver is able to measure carrier-phase observations from each satellite and use this carrier phase to perform RTK positioning.

Simulator In-Orbit Carrier-Phase RTK Test

High precision relative positioning is an important requirement for both Biarri and Garada. As a first step in validating that Namuru/Aquarius is a feasible platform for the provision of this functionality, two STR6560 GPS simulator low earth orbit scenarios were constructed[1], with the requirement that orbits were separated by no more than 10 km and that truth data for the orbital motion be generated and logged during the simulator run. RINEX measurement data was then acquired by running each scenario with a Namuru V2 receiver operating in 'Space' mode and converting the logged output to RINEX.

As before, RTKLIB was then used to perform the high precision relative positioning, although because no fixed base-station is available for this application, the 'rtkpost.exe' was run in 'moving-baseline' mode. This allowed the measured separation between the orbits as measured by the GPS receiver to be obtained. The last step in the process was to compare the measured separations with the true orbit separations as calculated from the simulator truth data files. These results are shown in the two plots shown in Fig. 4 and Fig 5, where it is obvious from Fig. 4 that there are some outliers in the positioning process. However, if the outliers are removed, then it can be seen that the underlying performance is actually quite good, with relative positioning errors of the order of 2 cm or so. The cause of the outliers is believed to be due to carrier phase reset events, which are believed to occur if the receiver loses phase lock. This problem will be addressed in a future revision of the Aquarius firmware.

[1] Semi-major axis of 7058 km, inclination of 98º, right-ascension of ascending node of 270º and mean-anomalies of 0º and 0.04º respectively.

Conclusions

This paper has described the hardware and firmware for a next generation of Namuru GPS receiver designed for the Biarri project, a cubesat project being undertaken in conjunction with DSTO and BAE Systems. Limitations of the previous generation Namuru V2 and the associated firmware were first discussed leading to a set of requirements for a Namuru V3.2 GPS receiver. New Aquarius firmware designed to overcome these limitations was described, as were the changes necessary to get the receiver to operate in a low earth orbit environment.

Finally, several test results showing the performance of the receiver under static, slow carrier phase RTK and in-orbit carrier phase RTK were presented. These show that the basic receiver functionality has been achieved, although there is scope for improvement and a need to eliminate some remaining firmware bugs. However, this is always the case for newly introduced products and given sufficient time, these problems can be eliminated.

Acknowledgements

Part of the Aquarius firmware development was funded by the ARC Discovery Grant DP1093982 (Preparing for the Next Generation Global Navigation Satellite System Era: Developing and Testing New User and Reference Station Receiver Designs) and the ASRP 'SAR Formation Flying' project. The development of the Namuru V3.2 hardware and the porting of the Aquarius firmware from the NIOS2 to the SmartFusion ARM Cortex M3 is funded by the Biarri project.

References

1. Nanosat Engineering Boeing, "Colony 2 Bus Payload Developer Guide: Version 1.2," Huntington Beach, CA: Boeing Phantom Works, 2011.
2. L. M. Sims, M. Riot, W. D. Vries, S. S. Olivier, A. Pertica, B. J. Bauman, D. Phillion, and S. Nikolaev, "Optical Payload for the STARE Mission," in *SPIE Defense and Security* Orlando, FL.
3. K. J. Parkinson, A. G. Dempster, P. J. Mumford, and C. Rizos, "FPGA based GPS receiver design considerations," *Journal of Global Positioning Systems,* vol. 5, pp. 70-75, 2006.
4. P. J. Mumford, K. J. Parkinson, and A. G. Dempster, "The Namuru Open GNSS Research Receiver," in *19'th Int. Tech. Meeting of the Satellite Division of the U.S. Inst. of Navigation* Fort Worth, TX, 2006, pp. 2847-2855.
5. F. Engel, G. Heiser, P. J. Mumford, K. J. Parkinson, and C. Rizos, "An Open GNSS Receiver Platform Architecture," in *GNSS 2004*, Sydney, 2004.
6. Zarlink-Semiconductor, "GP2015 GPS Receiver RF Front End," Zarlink Semiconductor, 2007.
7. Zarlink-Semiconductor, "GP2021 GPS 12-Channel Correlator," Zarlink Semiconductor, 2005, pp. 1 - 65.
8. E. P. Glennon and A. G. Dempster, "Delayed PIC for Postcorrelation Mitigation of Continuous Wave and Multiple Access Interference in GPS Receivers," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 47, October 2011.
9. A. Grillenberger, R. Rivas, M. Markgraf, P. J. Mumford, K. J. Parkinson, and C. Rizos, "The Namuru receiver as development platform for spaceborne GNSS applications," in *4th ESA Workshop on Satellite Navigation User Equipment Technologies* Noordwijk, The Netherlands, 2008.
10. O. Montenbruck, E. Gill, and M. Markgraf, "Phoenix-XNS - A Miniature Real Time Navigation System for Leo Satellites," in *3rd ESA Workshop on Satellite Navigation User Equipment Technologies, NAVITEC'2006* Noordwijk, 2006.
11. O. Montenbruck, M. Markgraf, M. Garcia, and A. Helm, "GPS for Microsatellites - Status and Perspectives," in *Small Satellites for Earth Observation*, R. Sandau, H.-P. Roser, and A. Valenzuela, Eds.: Springer Verlag, 2008, pp. 165-174.
12. K. J. Parkinson, P. J. Mumford, E. P. Glennon, N. C. Shivaramaiah, A. G. Dempster, and C. Rizos, "A Low cost Namuru V3 recever for Spacecraft operations," in *International Global Navigation Satellite Systems Society IGNSS Symposium 2011* University of New South Wales, 2011.

13. Y. S. Shmaliy, O. Ibarra-Manzano, L. Arceo-Miquel, and J. Munoz-Diaz, "Analysis of Sawtooth Noise in the Timing SynPaQ III GPS Sensor," *Sensors & Transducers Journal,* vol. 79, pp. 1151-1156, 2007.
14. E. Dupont, O. Lauzeral, R. Gaillard, and M. Olmos, "Radiation Results of the SER Test of Actel, Xilinx and Altera FPGA instances," 25 October 2004.
15. Mitel-Semiconductor, "GP2000 : GPS Chipset - Designers Guide : MS4395-2.3," Mitel Semiconductor, 1998.
16. GEC Plessey Semiconductors, "GPS Architect 12 Channel GPS Development System: DS4605-2.5," 1997.
17. E. P. Glennon, "Cross Correlation Mitigation for C/A Code GPS Receivers," PhD Thesis, School of Surveying and Spatial Information Systems, University of New South Wales, December 2009
18. R. Hatch, "The synergism of GPS code and carrier measurements," in *3rd International Geodetic Symposium on Satellite Doppler Positioning*, Las Cruces, NM, 1982, pp. 1213-1231.
19. P. Crawford, "The Dundee SGP4 code", 11 October 2007, Accessed 21 December 2011, http://www.sat.dundee.ac.uk/~psc/sgp4.html
20. Spirent Communications, "GSS6560 Multi-Channel GPS/SBAS Simulator Product Specification: Issue 3.01," 2004.
21. S. Takasu and A. Yasuda, "Development of the low-cost RTK-GPS receiver with an open source program RTKLIB," in *International Symposium on GPS/GNSS,*, Juju, Korea, 2009.