GOOSE – GNSS Receiver with an Open Software Interface

Matthias Overbeck*, Fabio Garzia*, Alexander Popugaev*, Oliver Kurz*, Frank Förster*, Wolfgang Felber*,
Ayse Sicramaz Ayaz◇, Sunjun Ko◇, Bernd Eissfeller◇,
**Fraunhofer Institute for Integrated Circuits IIS, Erlangen/Nuremberg, Germany*
◇*Universität der Bundeswehr München, Germany*

BIOGRAPHY

Matthias Overbeck received his Dipl.-Ing. (MSc) degree in Electrical Engineering from the University of Erlangen-Nuremberg, Germany, in 1999. Since September 1999 he is working at the Fraunhofer Institute for Integrated Circuits (IIS) in the Power Efficient Systems Department on the topic hardware development for global navigation satellite system (GNSS) receivers. Within this scope he developed digital baseband hardware like signal conditioning, correlators and microprocessor for field-programmable gate array (FPGA) and for application-specific integrated circuits (ASIC) for an assisted global positioning system (GPS)/European Geostationary Navigation Overlay Service (EGNOS) receiver for mobile applications. He gathered profound skills in the design of an application specific DSP for signal acquisition, tracking and the PVT solution for the same application. Moreover he has developed a rapid prototyping platform on FPGA basis for GPS/Galileo prototype receiver. Since 2009 he has taken over for Fraunhofer the technical leadership of the GSA project Galileo Receiver for Mass Market Applications in the Automotive Area (GAMMA-A). Since 2013 he is the Group Manager of Precise GNSS Receiver at Fraunhofer IIS.

Dr. Fabio Garzia received his MSc degree as Electronics Engineer in March 2005 from the University of Bologna, Italy. After his graduation, he worked for a few months at ARCES Labs in Bologna, first as VLSI Layout Engineer then as Digital Designer. In 2006 he moved to Tampere, Finland, where he started his PhD studies. His PhD Thesis focused on the design of a coarse-grain reconfigurable accelerator at RTL level, the development of programming tools, design of SoC platforms based on this accelerator, implementation on FPGA and mapping of some applications on it. He received his Dr.Tech. (PhD) Degree in December 2009. From January 2010 to April 2011, he received a research grant for the development of a high-level compiler for the reconfigurable accelerator. In October 2011 he moved to Germany, where he joined the Navigation Group, Power Efficient Systems Department at Fraunhofer IIS. His main task is the development of digital hardware and HW/SW interfaces for GNSS receivers targeting both ASIC and FPGA technologies. In May 2015 he received the title of Senior Engineer at Fraunhofer IIS.

Dr. Alexander E. Popugaev was born in Suzdal, Russian Federation, in 1981. He received the MSc degree (with honours) in radio engineering from the Vladimir State University, Russia, in 2004, and the Dr.-Ing. (PhD) degree (summa cum laude) from the Technical University of Ilmenau, Germany, in 2013. Since 2004, he has been with the Fraunhofer IIS, Erlangen, Germany. Currently, he is a senior engineer in the radio frequency (RF) and SatCom Systems Department. His current research activities focus on the design of customer-specific microstrip antennas, improving the efficiency of their design process and providing cost effective solutions. Dr. Popugaev has authored and co-authored several scientific papers and book chapters and has received several patents.

Oliver Kurz did his Bachelor of Physics at the University of Wuerzburg, Germany. After that he pursued a Master of Space Science and Technology at University of Wuerzburg, Germany, and Lulea, Sweden. He started to work at Fraunhofer IIS, Nuremberg, Germany, in 2008. Since then he is working in development of GNSS receiver technology, especially software. His experience in his present job and the responsibilities are system design, test management, supporting team infrastructure, software development and architecture for GNSS receivers, design and development of signal processing algorithms for satellite navigation systems, software related system integration in cooperation development and research projects.

Frank Förster received his Dipl.-Ing. degree in Electrical Engineering from the University of Erlangen-Nuremberg, Germany, in 2003. Since then, he is at the Fraunhofer IIS as a system design engineer. Currently he is involved in several navigation and communication projects where he is responsible for designing, evaluating and testing analog frontends for GNSS receivers in harsh signal environments.

Dr. Wolfgang Felber received his Dipl.-Ing. (MSc) de-

gree from Helmut Schmidt University - University of the Federal Armed Forces Hamburg in 2002 and his Dr.-Ing. (PhD) in 2006. He is working for the Fraunhofer IIS since 2009. From 2009 – 2011, he was a research scientist in the area of satellite communication in Ilmenau and was heavily involved in building up a test facility for “Satcom on the move” (SOTM) applications. This test facility is today one of the few approved GVF (Global VSAT Forum) test entities in the world for SOTM-terminals. From 2011-2014 he was Head of the Group “Satellite Communication” and since 2013 Deputy Head of Research Group “Wireless Distributed Systems / Digital Broadcasting” in Ilmenau. In 2014 he moved to Nuremberg and took over the Head of a long-standing department developing localisation technologies, mainly technologies for combined GPS/EGNOS/Galileo receivers for different applications. His department today is also working in inertial sensor systems and navigation, multi sensor systems and sensor fusion, high precision microwave localisation systems for sport applications, power management systems for communication transceivers and navigation receivers and low power communication modules for embedded systems. Before working for Fraunhofer he was with the Federal Armed Forces working as a Technical Officer at the Air Force in different technical units and positions.

Dr. Ayse Sicramaz Ayaz received the BSc degree in Electronics Engineering from Dokuz Eylül University, Turkey, the MSc degree in Information Technology from University of Erlangen-Nuremberg and PhD degree in 2013 at the Institute of Space Technology and Space Applications in University FAF. Major interests are signal processing implementations, synchronization, interference and mitigation algorithms in GNSS Receivers and Real Time Kinematic (RTK). Since 2004, she is a research associate in the Institute of Space Technology and Space Applications at the University FAF.

Dr. Sun-Jun Ko received his PhD in Electronics Engineering at Ajou University, S. Korea in 2006. He worked for R&D Institute of Samsung Electro-Mechanics from 2006 to 2009. Since 2010, he has worked with the institute of Space Technology and Space Application, University of FAF, Munich, Germany. He has a lot of experiences in aspect of receiver and next generation of communication devices, also published many papers on GPS/GNSS research area since 2000. His research interests are GNSS Signal processing, architecture of GNSS receivers, detection and mitigation of interference signal, and multi-sensor fusion for indoor navigation system and its implementation.

Prof. Bernd Eissfeller is Full Professor and Vice-Director of the Institute of Space Technology and Space Applications at the University FAF Munich. He is responsible for teaching and research in the field of navigation and signal processing. Till the end of 1993 he worked in industry as a project manager on the development of GPS/INS navigation systems. From 1994 - 2000 he was the head of the GNSS Laboratory and since 2000 full professor of Navigation at University FAF. He is author of more than 215 scientific and technical papers.

ABSTRACT

For the typical user of positioning and navigation applications, a precise GNSS receiver is a black box which delivers not well specified raw measurements with very reduced configuration possibilities. However, the processing of precise positions from such measurements needs a deeper insight about the way how these measurements were obtained (filtering etc.). In addition, precise positions are usually needed in critical environments like forests, fields and outdoor storage which require different kind of sensors to circumvent position disturbance caused by shadowing and reflections. In order to keep the precise position, the carrier phase solution needs to overcome short signal outages. The most common solution is to support the PLL with deeply coupled sensors. Unfortunately, this technique cannot be used with standard commercial receivers as there is no possibility to control the tracking loops from outside the receiver. In order to make this possible the objective of the GOOSE project (German acronym for ”GNSS Receiver with open software interface”) is to provide a flexible GNSS development platform for all kind of precise GNSS applications including an open software interface available to guarantee a transparent free and deep access to the receiver’s hardware.

INTRODUCTION

The GNSS receiver black box

Nowadays the GNSS receivers are available as chip set modules, OEM boards and full housing receiver products; all offering position, velocity and time component (PVT) and some additional raw measurements. Most of them are configurable in terms of update rates and some also in received signals and systems. Some of them even support the processing of additional sensors. But, what they all have in common is that the user has to arrange with the output measurement results not knowing how they where generated. The closed receiver systems don’t allow to run own software for machine steering on the receiver processor. Therefore such systems are always interfaced to additional devices like other receiver(s), processing engines and man-machine-interfaces (displays) spread over the drivers cabin. An even bigger problem is that it is not possible to feedback sensor information to stabilize the receiver tracking loops in harsh environment. So the possibility of using nowadays receiver in several applications is very limited.

State of The Art

The GOOSE platform provides the flexibility of an academic receiver setup combined with the performance of professional OEM boards. Therefore in the next two subsections the state of the art of academic and OEM receivers is described and compared to GOOSE.

Academic Open Receiver Approaches

The „University of New South Wales“ provides the Namuru receivers which use components off the shelf for the frontend, combined with an FPGA as the hardware base. Most of them use the GP2015 Zarlink chip for GPS L1 as a frontend base and the soft-processor NIOSII on Altera FPGA [1]. Together with the Auquarius software, an experimental single frequency GNSS receiver can be realized [2]. In the newest versions of the hardware dual-band frontends are provided. The focus has been to be robust enough for low earth orbit operation and less on high processing power nor on multi signals and systems.

A similar approach is used by the Tampere University of Technology using an Altera FPGA with the NIOSII soft core [3]. Three different L1-frequency front-ends from Atmel, Sige and NEMERIX were used with their setup. Again the limiting factors are the number of channels. There was only space for 16 channels left which is sufficient for one system and one signal (GPS L1) but starts already to get pretty too little if Galileo is added or several frequencies and signals. The receiver development was part of the FP7 GRAMMAR project [4].

Also a couple of software defined radio (SDR) approaches can be found. They are all limited by the amount of data which can be processed in realtime. Allowing higher bandwidths with higher sampling rates reduces the number of channels and with that the number of signals and systems which can be processed.

Professional OEM Boards

Despite of the academic flexible receivers every market player in the GNSS receiver sector offers several OEM boards for integration in own systems. These OEM boards are the before mentioned black boxes, which offer high performance raw-data and position precision but neither give an insight as to how these measurements are generated nor allow to run own applications on the receiver processor. Two high-end example OEM boards from well known manufacturers are described.

The TR-G3T is one of the highend OEM boards from Javad and provides from revision 5 on three signal bands for reception and tracking of GPS L1/L2/L2C/L5, Galileo E1/E5 AltBOC, and GLObal NAvigation Satellite System (GLONASS) L1/L2/L3 [5]. The digital signal processing is executed in the TRIUMPH® Chip offers up to 216 channels split in 72 standard channels with 5 correlators and 48 channels for combined data pilot tracking with 10 correlators as well as 48 memory code channels with again 5 correlators for Galileo [6]. A fast acquisition engine is used with 100k correlator equivalents. The given precision of 10 cm code phase and 1 mm carrier phase precision is state of the art.

The OEM638 from Novatel is the flagship of their OEM products. It also provides state of the art precision [7]. With up to 240 channels it is able to track GPS L1/L2/L2C/L5, GLONASS L1/L2, Galileo E1/E5a/E5b/AltBOC, BeiDou B1/B2, SBAS and QZSS . The signal processing from intermediate frequency is done by two MINOS6® ASICs [8]. As a triple frequency frontend is used not all signals are available at the same time.

GOOSE Approach

In this paper we present the GOOSE development chain to overcome the limitation of state of the art receivers and take GNSS researchers with their development work from idea to demonstrator, assisting them in finding the right receiver configuration. The limitations on academic side are the number of supported systems, signals and channels and on the professional side the black-box concept. The requirements to overcome these limitations is a flexible hardware supporting all GNSS bands, a fast hardware signal processing with open access, high processing performance and an open software interface. The steps from demonstrator to a product after successful demonstration with GOOSE can be kept small with low risk of failure.

This paper is divided into the following sections. In the section GNSS Receiver with an Open Software Interface the different components which build up the GOOSE platform software are described. This is the new L-Band antenna, the triple band frontend, the baseband hardware implementation on FPGA, the software with the open interface and the GOOSE development chain.

The ”Preliminary Receiver Evaluation” section gives a short status about current measurements taken from the GOOSE receiver.

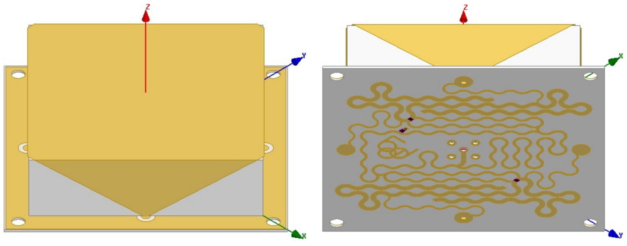
A short sum up of the project goals and the possibilities coming with the GOOSE platform is given in section ”Conlusion”.

GNSS RECEIVER WITH AN OPEN SOFTWARE INTERFACE

Antenna

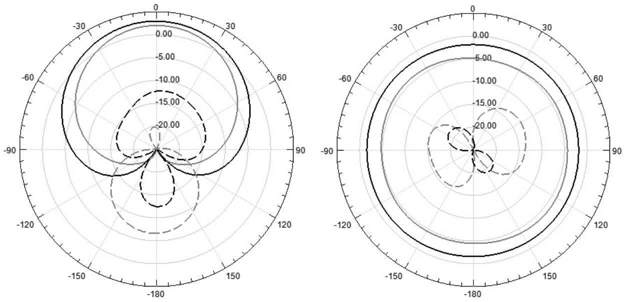
The GOOSE hardware platform supports two different GNSS antenna inputs. The first one is for a compact embedded antenna. The second one can be used for an external high-performance GNSS antenna. As the GOOSE antenna should provide a possibility for system extendability, both antennas have been designed to operate for all existing and planned satellite navigation frequencies in L-band.

The active stand-alone GNSS antenna has been already described in [9]. Amongst others, the distinctive feature of the design is a broadband square radiator with four triangular legs and a miniaturized four-point circular polarization feed network composed of quarter-circular rings [10]. Through replacement of the antenna electronic to the GOOSE hardware platform and some modifications of the radiator (filling with a dielectric material) as well as the feed network, a more compact design has been developed – see figure 1. The miniaturized antenna requires a certain ground plane size (about $100 \times 100\ \mathrm{mm}^2$) to achieve acceptable performance in terms of antenna gain, polarization purity and impedance matching within the specified frequency range. As such, the metalization of the receiver board is used. The radiation pattern is shown in figure 2.



(a) Radiator (b) Feed network

Figure 1. Embedded GOOSE antenna ($60 \times 60\ \mathrm{mm}^2$)



(a) Vertical cut: azim. 45° (b) Horizontal cut: elev. 30°

Figure 2. Radiation pattern of the embedded GOOSE antenna (passive gain in dBic): RHCP (solid line) and LHCP (dashed line) at 1.16 GHz (black) and 1.61 GHz (grey)

Frontend

The GOOSE front end allows the reception of three separate GNSS bands with bandwidths between 40 and 68 MHz. It is composed of discrete components and includes a low noise amplifier (LNA), band pass filters, a configurable RF integrated circuit (IC) with a local oscillator (LO) and an analog-to-digital converter (ADC). The first reception channel covers in standard configuration the signals of GPS L1, Galileo E1, GLONASS G1 and Beidou B1, the second channel receives the L2/L2C-band and GLONASS G2 and the third path obtains E5, E5a, E5b, L5 and B2.

The frontend board can be found as one major part of figure 6. The block diagram for one reception path is illustrated in figure 3. The first LNA is used to guarantee a low overall noise figure (NF). A three-way splitter distributes the amplified antenna input signal to the dedicated RF bandpass filter which attenuates out-of-band interferences and selects the reception band. Dielectric RF filters are used to obtain an excellent group delay behavior. A commercial off the shelf I/Q downconverter RF IC is used providing a zero-IF or low-IF architecture depending on its LO setting. The LO (and therefore the resulting intermediate frequency (IF)), the amplifier gain and the anti-aliasing low-pass filter bandwidth can be configured using SPI and I2C interfaces. The setting of these parameters is performed by the baseband board. A control loop is implemented and sets the variable gain amplifier according to the ADC-level.

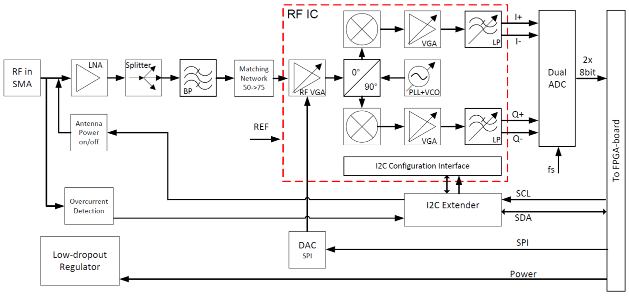


Figure 3. Frontend block diagram of one reception path

The RF IC budget plan concerning its gain, noise figure (NF) and input intercept point 3rd order (IIP3) is given in table 1 for the L1 configuration. Since the RF IC has a 75 input impedance, a broadband resistive matching network is used for interconnection with the RF filter, adding some implementation loss. The frontend is designed to be used with an active antenna. The noise figure is then determined by the antenna. The DC power supply for the active antenna is controlled by the baseband board. Three dual-channel ADCs run at 81 MSPS sampling rate and 8 bit resolution.

A dedicated clock generation and distribution chip is used to coherently derive all the frequencies required for the RF IC, the ADCs, and the FPGA. The reference clock comes from a 10MHz temperature controlled crystal oscillator (TCXO). The board provides a buffered clock output. An external reference input is also available where clocks with a higher performance can be attached. The TCXO has an Allen Deviation of 0.1ppb which shows a good tradeoff between cost and performance.

At the moment an alternative front end is brought on the way which supports 4 signal bands from 400 MHz to 3.7 GHz and will be also fully supported by the GOOSE platform.

	LNA	Splitter	RF filter	50→ 75R	RF IC	overall
GAIN [dB]	14	-6	-1.2	-4	75	77.8
NF [dB]	1.2	6	1.2	4	8	6.6
IIP3 [dBm]	25	50	50	50	-40	-39.5

Table 1. Frontend budget plan

Baseband

The baseband board consists of a customized PCB connected to the front end through a Samtec connector (see figure 6). The core of the baseband board is a XC7K410T FPGA where a dedicated GNSS HW subsystem is implemented. By default it is planned to provide digital signal conditioning and hardware channels for tracking of 90 satellite signals using 900 correlators. The versatile tracking channels can be mapped to any combination of GPS, Galileo and GLONASS signals. On a lower level the signal can be evaluated with the help of raw signal recording which can for example be used for software based acquisition, interference detection, characterization and mitigation purposes. The tracking channels and the complete hardware are controlled via Peripheral Component Interconnect Express (PCIe) either by a single-board computer (SBC) equipped with an ARM® A9 processor or a desktop computer. In order to allow that, the board is equipped with a standard SBC connector and an optional PCIe riser card is provided as an adapter between the SBC connector and the PCIe connector of a PC motherboard. This choice provides additional flexibility on the user environment, since the system can be inserted in a common PC for laboratory tests or used as standalone board for on-the-field tests. Both approaches offer enough processing power to run complex user applications and algorithms.

FPGA baseband subsystem

The architecture of the FPGA baseband subsystem is depicted in figure 4.

The baseband subsystem provides the core functionalities required in a GNSS receiver using dedicated hardware modules controlled through AMBA/AXI ports. A Xilinx core provides the additional interface between the AXI interconnections and the PCIe port which is used to communicate with the SBC or the PC. The physical PCIe is fast enough to handle the amount of data and has the flexibility to use the GOOSE hardware either in a PC or together with a single board computer (SBC, SMARC-sAMX6i) as an embedded platform.

One of the dedicated HW modules instantiated in the system is a FFT-based acquisition core, which allows the detection of L1/E1 satellites, their Doppler and code offset. A typical acquisition procedure is based on a iteration over all the possible satellites and Doppler bins. For each try, the acquisition module returns a flag and, in case of success, the peak position and the peak size. The peak size can be used to make a decision over the Doppler, the peak position provides a measure of the code offset. Here the user has already room for algorithmic exploration, i.e., which Dopplers should be tested (large Doppler bin for speed, small Doppler bins for precision) plus the possibility to have a variable amount of coherent or incoherent accumulations. In addition to the acquisition module, the baseband is equipped with dedicated correlators to track the detected satellite signals using FLL/DLL/PLL implemented in SW. The current target is to have up to 90 5-tap tracking channels, i.e., each one equipped with 5 complex correlators (VE/E/P/L/VL), which can be mapped to any combination of GPS, Galileo and GLONASS signals and different frequency bands. Here the user has a direct access to the correlator control interface, the integrate-and-dump values needed for the loop closure and the measurement interface (code phase and carrier phase). The user can specify the integration time and the measurement rate. Increased flexibility is guaranteed by the fact that the loops are closed in SW by the SBC or the PC.

Also, an FPGA is used as target platform. This allows on-the-field deployment of different architectural configurations tailored to specific user needs. For example, the user can decide to remove the acquisition module and use only the correlators to perform a Tong-search acquisition and get a large amount of channels. Or the user can choose between different channel configurations: instead of the standard one, a configuration with 14 32-tap channels for advanced tracking features (e.g. multipath detection) or a configuration with up to 150 3-tap channels to support only BPSK signals.

The main software building blocks, like tracking loop control, measurement generation and PVT are all not hard-

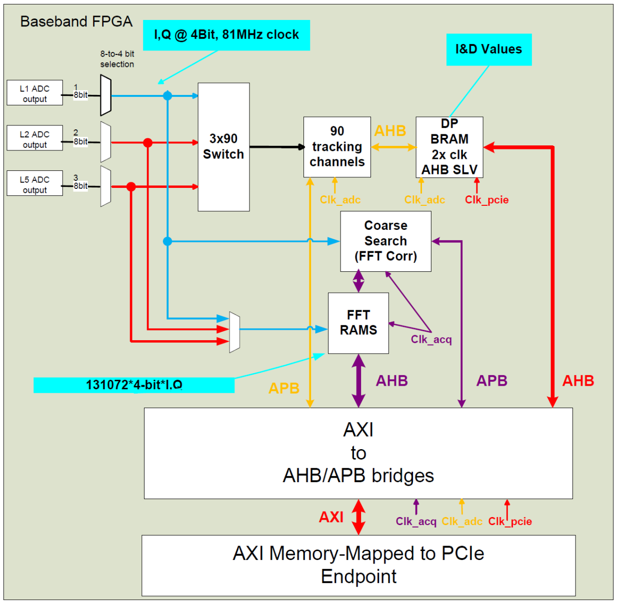


Figure 4. FPGA baseband block diagram

coded in the baseband component but separated over the flexible interface. They are available as replaceable software components that can be used as-is in the user application or replaced by the customized implementations.

Open software interface

In the field of GNSS there exist few standards for data exchange and a multitude of proprietary protocols specific to individual receiver manufacturers. For raw measurements the Receiver Independent Exchange Format (RINEX) text protocol can be considered standard. The format is text based and designed for post-processing use. It cannot be used straightforward for real-time data exchange between a receiver baseband part providing the measurements and a consumer of these measurements, e.g. a standard PVT or coupling for sensor fusion using an Intertial Navigation System (INS). Another widely used protocol used in satellite navigation devices is *NMEA 0183*. It is simple to process however very limited to deliver measurements of a receiver. For higher demands of bandwidth, e.g. when exchanging raw measurements like carrier phase or code range measurements or even undecoded subframe content, it is unsuitable. For RTK applications as also considered in the project GOOSE other approaches are commonly followed but most of them are tuned towards a special use case and therefore limited.

The goal of Open GNSS Receiver Protocol (OGRP), as it is defined as part of the development undergone in the project GOOSE, is to offer a well-defined and self-describing format in terms of available measurements and units while being vendor neutral. It was also required to represent all data of current GNSS formats also in OGRP. One requirement in design of the protocol was to allow archiving messages as well as allowing real-time exchange of data between different components of a system, e.g. a low-level receiver outputting measurements and a high-level consumer of the measurements. The format also fulfills the demand to be easily readable and simple while still providing flexibility for future enhancements and vendor specific data. To fulfill this specification a protocol based on JavaScript Object Notation (JSON) was designed. Many libraries for implementation in various programming languages exist to parse the data. In comparison to XML there is less keyword overhead and an even easier specification. OGRP is meant to be extend able and future-proof rather than optimized regarding performance for a specific case. The main messages are already defined and some of them can be extended by vendor specific properties. Messages that are not defined can be easily defined into the common scheme based on custom requirements. Modern processing platforms are powerful enough to handle OGRP in real-time with insignificant processor load. Though it is a text based protocol, it can still be used efficiently by adding a layer of compression around it, e.g. for communication over a physical link with limited bandwidth. The transport mechanism, e.g. TCP, RS232, etc., is not part of the OGRP definition. Any means of exchanging serial data can be used. The simple *Line Delimited JSON* standard is proposed to use in stream protocols.

The GOOSE receiver comes already with example implementation of the signal processing GPS L1 and L5 as well as for Galileo E1 and E5a. Further signals like GPS L2C and Galileo E5 AltBOC processing are currently under development in followup projects based on the GOOSE platform.

These software modules are interconnected via OGRP. A proposal for standardization exists and is available in a public github repository [11].

Development chain

The development chain is setup by different hardware variants and a software development environment supporting these.

Different hardware variants

The GOOSE hardware platform is available as a PCIe slot card setup as well as a portable version which uses a SBC as processing platform. This is possible because of using the same physical interface PCIe between the software part and the hardware of GOOSE for both hardware variants.

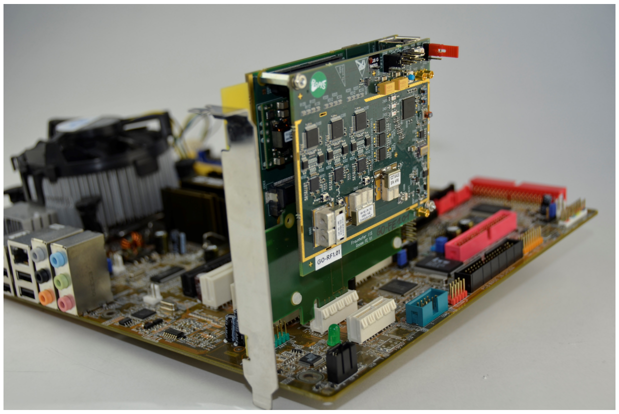


Figure 5. PC based receiver hardware

The PC based variant shown in figure 5 builds the starting point for the development. The developer has high processing power and an easy to use, comfortable Ubuntu linux platform. With the PC based platform tests with signal generator or stationary antennas can be conducted. The PC based variant can probably not setup enough mobile for geodetic field tests but could be used in the PC racks in the boot of cars prepared for development.

The embedded variant marks the technology readiness level (TRL) 6 which means it is foreseen to conduct field tests with this platform (see figure 6). The step from PC based platform to embedded is kept small as the development environment (see next section) fully supports both platforms. For a later product development the platform has to be optimized in form factor and power consumption. In that way GOOSE provides a complete development chain supporting developers from a PC based solution to a field test ready embedded professional receiver.

Software development environment

The goals of the development chain for GOOSE users are to have minimal effort to use a PC based GNSS platform as well as do a small step to go to an embedded platform based on a compatible architecture. The SBC as used as part of the embedded platform is provided with *Linaro*, an Ubuntu based GNU/Linux distribution for ARM architecture, current version 12.04 LTS precise pangolin. To be able to develop applications for the target platform OS a development package is provided which should make this process as easy as possible. The proposed PC based development environment is Ubuntu for x86_64.

EXAMPLE RTK APPLICATION SOFTWARE

In order to provide precise position estimates, an RTK unit is implemented and is evaluated in post processing which operates on the platform developed. Comparing with code-range based GNSS receivers, an RTK system has more complicated architecture in aspect of the algorithms, communication, infrastructures, and etc. The followings are the key factors for RTK system development.

- The carrier-phase measurements based on dual-frequency
- Integer ambiguity resolution techniques
- RTK reference stations, i.e. International GNSS Service (IGS) sites
- Communication link between reference and rover

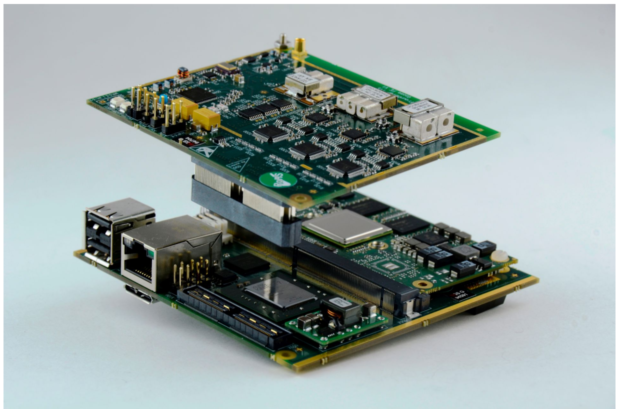


Figure 6. Embedded receiver hardware

Carrier-phase measurements can be the key factor to achieve more accurate positioning estimates, since it provides more fine resolution and more accurate ranges than pseudorange measurements. However it has integer cycle ambiguities which are not able to be estimated due to the increase of unknown parameters when generating carrier-phase ranges between satellites and receivers. In order to solve the problem, several integer ambiguity resolution techniques have been studied and Least-squares Ambiguity Decorrelation Adjustment (LAMBDA) technique is well known as one of the best integer ambiguity estimators. In this paper LAMBDA has been implemented to provide the best RTK performance through finding optimal ambiguity solutions. In addition, the relative positioning scheme is also fundamental for RTK system, since it reduces the unknown parameters required to estimate through removing common biases between measurements from different locations. Although several differencing methods can be utilized corresponding to Linear Combination models, such as ionosphere free model and geometry free model, Double Difference (DD) is generally used for removing clock bias and ionosphere delay. Based on such a requirement, it needs the reference station and the appropriate communication link to deliver raw measurements to users. In general, Radio Technical Commission for Maritime Services (RTCM) standard is widely used, and users can easily receive it from IGS sites. In this paper, RTK reference station which supports RTCM standard via User Datagram Protocol (UDP) based on wired and wireless connection is implemented.

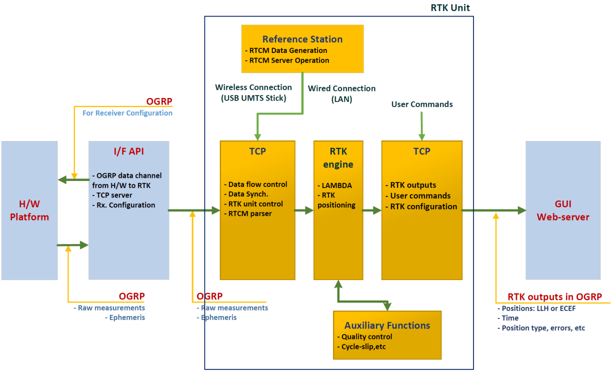


Figure 7. Architecture of the implemented RTK system

Figure 7 shows the RTK system architecture implemented following with generic RTK architecture. It has several components, such as receiver, RTK engine and auxiliary functions, Interface APIs, and GUI web-server. The followings are descriptions of each operation.

- Receiver
  - Generation of raw measurements and OGRP

  - Interface between hardware and RTK unit
  - Receiver configuration and initialization
- RTK unit
  - RTK engine: operation in integer ambiguity resolution and estimation in precise positions
  - TCP: Interface module to receive OGRP from receiver. Also it supports RTK outputs in OGRP to web-server
  - Reference station: RTCM standard generation and transferring it to RTK engine via wired or wireless connection (UDP)
  - Auxiliary functions: imrpoving quality of integer ambiguity estimates and preventing abnormal data
- GUI web-server
  - User interface and GUI for RTK outputs
  - Interface between RTK unit and server via TCP

As mentioned in the section above, OGRP based on JSON provides the flexible data protocol for GNSS receivers, which makes it available to easily generate a user defined protocol. The appropriate data protocol has been designed to include all raw measurements, ephemerides, clocks, and initial positions. The RTK unit defines two types of OGRP as the input and output data corresponding to data flow and its requirement. The physical links are used in wired TCP connection via LAN.

Reference station tracks all visible satellite signals and it provides raw measurements having the highest accuracy. Based on such a high performance, it generates and transmits RTCM standard data including raw measurements, antenna information, and data relating to the station. In general, RTCM standard 2.3 and 3.1 version are widely used, and transferred to RTK user through Ethernet-link. The implemented reference station provides the same functions as generic reference stations and RTCM standard 2.3 and 3.1 version are also supported.

When RTK unit processes RTCM standard, time synchronization between RTCM standard and OGRP is quite important because RTK positioning performance becomes worse and integer ambiguity searching would be failed due to data latency. Therefore the data latency should be maintained within the reasonable time range from several milliseconds to seconds. To cope with a large data latency, RTK unit stacks several OGRP messages (10 seconds) to find the right data having the same measurement time. After synchronization, raw measurements can be used to calculate in DD. UDP link is used for RTCM standard connection using LAN and WLAN via UMTS device.

As RTK engine is the core part in RTK the unit, it provides precise position estimates, integer ambiguities, precise time, and others relating to navigational information. LAMBDA estimates the optimal integer ambiguity solutions, and RTK position estimator fixes precise positions having the centimeter-level accuracy. Such outputs are transmitting to web-server which tracks and demonstrates RTK positioning outputs in real-time and evaluates its performance. Its communication link is also based on TCP connection.

All of RTK unit modules described in this paper are implemented in software APIs and individual programs to easily perform unit-tests and to guarantee the performance of each module without collisions between software modules. Their performance is evaluated in post processing and file test.

Reference Station

The Reference Station (RS) tracks all visible satellite signals and provides measurements of the highest accuracy. It is capable to generate output RTCM standard V2.3 / V3.1 delivering it to RTK rover through Ethernet port. Corrections are broadcasted from the reference receiver placed at a known location. There is also the possibility to store the messages within a log file.

The data link between the RTK rover and the RS is critical since in case of a correction failure or latency the system performance may be degraded. Therefore, a high data rate UMTS network is used to receive data corrections from reference station over UDP.

PRELIMINARY RECEIVER EVALUATION

In a first development stage the PC based platform is setup in a laboratory connected to a roof antenna receiving a real GNSS signal from satellites. This setup was verified to successfully track GPS L1 C/A + L5I. Next step in the verification is to verify tracking of Galileo E1B + E5aI using the Galileo FOC satellites.

Example output of a measurement message in OGRP format showing the measurements of two channels tracking GPS satellite nr. 14, a Block IIR(M) satellite. One channel is successfully tracking GPS L1 C/A, the other one is searching for the L5 signal, which is actually not present on this satellite:

```
{"ch_meas":[{"carrier_phase":198694980.7956696,"
    ch_nr":3,"channel_state":"SYNCED","doppler
    ":-2023.860188201070,"gnss":"GPS","locktime
    ":4660.0,"pseudorange":22571871.58366107,"
    sat_id":14,"signal_type":"L1CA","snr
    ":48.07287465853545},{"carrier_phase
    ":34179599168.21297,"ch_nr":43,"
    channel_state":"SEARCHING","code_phase
    ":0.0,"doppler":1894.189696758986,"gnss":"
    GPS","locktime":0.0,"pseudorange":-1.0,"
    sat_id":14,"signal_type":"L5I","snr
```

```
":31.21265038674681}],"id":"measurement","
protocol":"OGRP1","sw_version":32768,"
time_status":"FINESTEERING","timestamp
":1116253497.0}
```

As OGRP is based on JSON an evaluation is straight forward based on the easily parseable message content. Figure 8 shows a “Code minus double carrier” (CMDC) evaluation of a one-hour long measurement based on a simple evaluation script reading in JSON using a standard library in python programming language, calculating the CMDC from the data included in OGRP for one satellite signal and plotting the values together with statistical parameters.

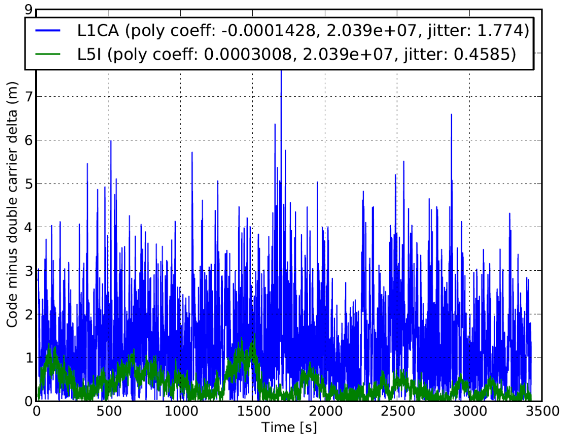


Figure 8. Evaluation of OGRP data yielding CMDC from a preliminary roof antenna setup showing a mean code jitter of 1.8 m (GPS L1 C/A) / 0.5 m (GPS L5I)

For this measurement the Phase Locked Loop (PLL) filter order has been 3 with a bandwidth of 10 Hz and for the Delay Locked Loop (DLL) filter the order has been 2 with a bandwidth of 1 Hz. For L1 the early late spacing was approximately 0.4 and for L5 1.0 chips.

CONCLUSION

The GOOSE hardware platform provides to the GNSS community a development chain from experimental PCIe slot card to a professional embedded GNSS receiver. The GOOSE platform can be seen as a hardware-assisted software receiver where computational complex methods are implemented on digital FPGA hardware whereas algorithms can be developed and implemented on receiver side on a user friendly GNU/Linux system. A transparent access to the hardware is made available via the Open GNSS Receiver Protocol which gives deep access to the hardware control and enables deeply coupling of inertial sensors and optimized precise positioning solutions. It is therefore targeted at researchers, software developers and algorithm experts to build up new methods and applications in the GNSS area. At the end of the project 20 GOOSE platforms will be available for selected researchers for free. The main benefits for potential product developers are an improved development process for GNSS receiver firmware, the possibility to embed application-specific software on the receiver, an access to all potentially relevant data for an improved position solution based on open white box approach and the enabling of deeply coupling of inertial sensors.

ACKNOWLEDGMENT

The GOOSE project is funded by the ”Bundesministerium für Wirtschaft und Energie” (German Federal Ministry for Economic Affairs and Energy).

REFERENCES

[1] N. Shivaramaiah, J. Wu, J. W. Cheong, M. Choudhury, and K. Parkinson, “Annex 5. Developing a Satellite Navigation Receiver for the Space Mission.” online, June 2013. `http://www.garada.unsw.edu.au/Final%20Report/Annex%205.%20Developing%20a%20Satellite%20Navigation%20Receiver%20for%20the%20Space%20Mission.pdf`.

[2] K. P. Eamonn P. Glennon, Peter J. Mumford, “Aquarius Firmware for UNSW Namuru GPS Receivers.” online, November 2011.

[3] T. Paakki, J. Raasakka, F. Della Rosa, H. Hurskainen, and J. Nurmi, “Tutgnss university based hardware/software gnss receiver for research purposes,” in *Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS), 2010*, pp. 1–6, Oct 2010.

[4] S. Sand, “Galileo Ready Advanced Mass MArket Receiver, Final Report,” tech. rep., DLR, 2011.

[5] Javad, “TR-G3T.” online, October 2012. `http://www.javad.com/downloads/javadgnss/sheets/TR-G3T_Rev.5_Datasheet.pdf`.

[6] Javad, “Triumph Chip.” online, 2008. `http://www.javad.com/downloads/javadgnss/sheets/TRIUMPH.pdf`.

[7] Novatel, “OEM638.” online, March 2015. `http://www.novatel.com/assets/Documents/Papers/OEM638-PS-D17916.pdf`.

[8] Novatel, “OEM6® Family; Installation and Operation User Manual.” online, January 2015. `http://www.novatel.com/assets/Documents/Manuals/om-20000128.pdf`.

[9] A. E. Popugaev and R. Wansch, “Multiband GNSS antenna,” *A. Heuberger, G. Elst, R. Hanke (Ed.): Microelectronic Systems: Circuits, Systems and Applications, Springer, Berlin/Heidelberg*, 2011.

[10] A. E. Popugaev, *Miniaturisierte Mikrostreifenleitungs-Schaltungen bestehend aus zusammengesetzten Viertelkreisringen.* PhD thesis, Technical University Ilmenau, Germany, 2014.

[11] “OGRP - the Open Gnss Receiver Protocol,” 2014. `https://github.com/Fraunhofer-IIS/ogrp`.