OVERVIEW

The u-Nav microelectronics' OrionTM software is a powerful and versatile Multi-Mode GPS navigation software designed to support the u-Nav uN9018 GPS Receiver chipset solution across multiple frequency plans (see Table1).

Reference (MHz)	Application
13.0	GSM
16.3574	Standard
16.3676	Standard
19.2	CDMA
26.0	GSM
32.7452	Standard

Table 1: Orion Frequency Plans

The software is responsible for all GPS functions such as signal acquisition, tracking, data extraction, and GPS navigation. The navigation data is transferred from the

u-Nav Baseband processor using proprietary u-Nav Binary Protocol (UBP) or NMEA messages over UART serial interfaces. The application software can communicate back to the navigation software using NMEA or the UBP.

An Orion Software Development Kit (SDK) is also available, providing developers the freedom to customize and add functional blocks to meet application specific requirements.

Features and Performance

- Multi-Mode GPS software supports:
- Autonomous GPS
  Assisted conventional
  A-GPS Single Fix (3GPP, 3GPP2, IS-801,
  TIA-916)
- High sensitivity solution capable of:
  Autonomous data decoding below -142dBm
  Tracking at -151dBm
- Modular software architecture easily integrates into existing systems
- Software Development Kit (SDK) provides flexibility for software customization

Accuracy

Position (2DRMS)	<3 m
Velocity (RMS)	0.2m/s

TTFF

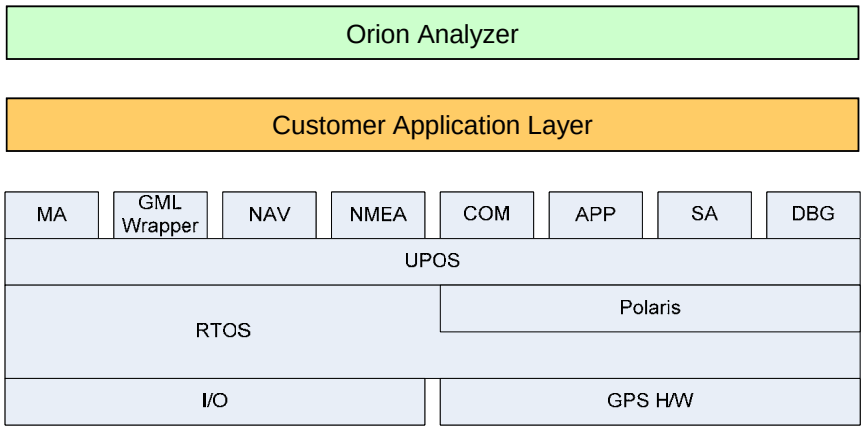
Fast Start	3 sec
Hot Start	8 sec
Cold Start	42 sec

SENSITIVITY*

Single Fix A-GPS	-151 dBm
Cold Start Acquisition	-140 dBm
Tracking	-151 dBm

** Specified at input of the receiver*

ORION SOFTWARE STACK



MULTI-MODE GPS NAVIGATION

Orion software supports autonomous conventional, assisted conventional, and Single-Fix AGPS modes. In autonomous conventional mode, the Orion software computes the receiver's current position utilizing measured pseudo-ranges, satellite ephemeris, clock correction, GPS time, and the receiver's last known position.

In assisted conventional and Single-Fix A-GPS modes, the Orion software utilizes externally provided assistance data to reduce Time to First Fix (TTFF), improve initial position accuracy, and increase sensitivity. Actual performance will depend on the quality of the external assistance data provided. For 3GPP, 3GPP2, and IS-801 compliant assistance data, the Orion software exceeds the specified performance requirements for each of those standards.

SATELLITE ACQUISITION AND TRACKING

The OrionTM navigation software features the u-Nav GPS Measurement Layer (GML™) that interfaces to the u-Nav baseband and RFIC hardware.

The GML acquisition algorithms efficiently scan GPS signals by utilizing the u-Nav QwikLock™ search engine, which simultaneously searches up to four frequency bins to improve initial satellite search time with no aiding information.

The GML tracking loop manages an array of twelve independent tracking channels and quickly adapts to changes in signal strength and other dynamic conditions found in harsh urban canyons.

WAAS / EGNOS

Orion supports WAAS and EGNOS signal tracking and message decoding. The message data provides satellite integrity information and correction for signal propagation delays due to the ionosphere, as well as corrections for GPS satellite orbit and clock drift. The ionosphere correction data improves the precision of the measured pseudoranges in the navigation layer, and the satellite integrity information is taken into account by the Orion navigation core while computing the position and the accuracy of the satellite signals.

SOFTWARE ARCHITECTURE

The architecture of the Orion software (Figure 1) is based on an information flow model that consists of message exchange between individual subsystems.

Each subsystem has a standard interface and contains at least one task. It can also include function libraries and interrupt handlers. The description of Orion subsystems is shown in Table 2.

The subsystem can only send and receive messages to other subsystems and the embedded RTOS through a centralized Messaging Agent (MA). The MA forwards the messages to the subsystem and provides valid and up-to-date information to all subsystems based on subscription or on demand by other subsystems.

Standard Subsystem - The Orion software architecture (Figure 1) is comprised of standard subsystems that are used in both conventional and assisted GPS navigation modes. The descriptions of the modules are listed in Table 1.

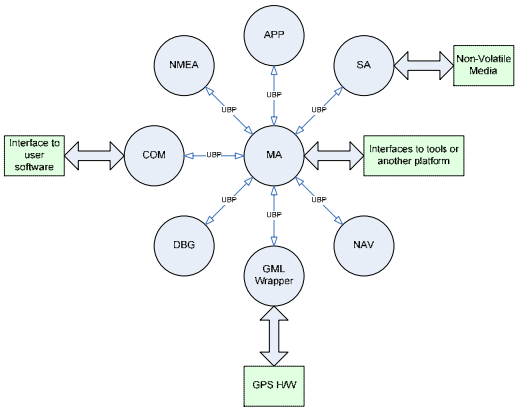


Figure 1: Orion System Architecture

APPLICATION PROFILE CONFIGURATION

The Orion software allows the user to send UBP or NMEA Application Profile commands that sets, stores, and retrieves receiver configuration information in the system. The Application Profile simplifies steps needed to configure and control the receiver features and navigation parameters.

Formatted: Font: (Default) Arial, 10 pt, Bold

Subsystem Module	Description
COM	Output navigation information and handles all communication between the HOST and Orion. This module handles NMEA formatted data from the NMEA subsystem, UBP data received from the NAV subsystem, and AGPS protocol data received from the AGPS subsystem.
NAV	Responsible for all calculations required to produce navigation fixes. The NAV subsystem can use external assist data from the AA or RAIM subsystem for better accuracy. The NAV produces Doppler estimates for GML, ephemeris and almanac data to SA for storage and later use, and assistance data (PVT) for the AA module.
GML	Produces measured code phases and Doppler to NAV subsystem and inputs aiding data from AA and Doppler estimates from NAV.
Time Agent (TA)	Manages both time (GPS time, UTC time) and timing related actions. TA is aware of available timing resources, including timing assistance.
NMEA	Convert message strings between u-Nav binary protocol and NMEA standard ASCII protocol.
Application Profile (APP PROF)	Handles application dependent parameters and distributes to other subsystem. Possible pre-defined profiles could be: vehicle use, pedestrian use, indoor use
Storage Agent (SA)	Responsible for storing and retrieving information considered non-volatile. The SA subsystem is aware of available storage subsystems, and hides media-specifics from the users.
DBG	Handles debugging-related messages. Features include: - Execution time profiler - Direct memory access - Stack usage reporting

Table 2: Orion Subsystem Descriptions

Command Type	Description
Communication	Defines communication protocol, UART port selection, message interval, port speed, message mask
Control	Issues Stop and Start commands and Sleep modes. Start navigation options are: - AUTO_START - COLD_START - WARM_START - HOT_START
Datum	Selects datum, configures offset from default datum (WGS84)
User Customization	Sets time zone
Debugger	Customizes interval for sending debug messages
Version (Read only)	Version of Orion software and u-Nav GPS hardware
Master Clock (Read Only)	Current frequency plan

Table 3: Application Profile Command Type

Additional information of the Application Profile features and commands can be found in the u-Nav Orion User's Manual.

NMEA Messages

Orion produces messages in accordance with NMEA 0183 v3.01 standard (http://www.nmea.org). The NMEA message data consist of ASCII characters based on the following format:

$GP<message id>,<data field>,<data field>,,,
..*<checksum><CR><LF>

The Orion software supports the NMEA messages listed in Table 4.

Message ID	Description
GGA	GPS fix data
GLL	Geographic position, longitude and latitude
GSA	DOP and active satellites
GSV	Satellites in view
RMC	Recommended minimum specific GNSS Data
VTG	Course over ground and ground speed
ZDA DTM	Time and date

Table 4: Orion™ NMEA Message Description

NMEA Commands

In addition to the NMEA messages listed in Table 4, u-Nav utilizes custom NMEA data and command messages based on the following format:

$PUNV, <command>,<data field>,<data field>,,,
*<checksum> <CR><LF>

Detail information of the NMEA messages and commands can be found in the u-Nav Orion NMEA User Manual document.

UBP PROTOCOL

The Orion u-Nav Binary Protocol (UBP) is used to communicate with external devices as well as internal communication between subsystem modules.

Table 5 provides examples of UBP messages types used in the Orion software. Detailed information of the protocol can be found in the u-Nav Proprietary Binary Protocol specification document.

UBP Message	Function
Registration	Registration of the module with the Messaging Agent. Normally used in the start up of system.
Data	GPS tracking status, satellite information, and navigation data.
Assistance	Assistance data.
Command	Start and Stop navigation commands sent by host. Start navigation options are: - AUTO_START - COLD_START - WARM_START - HOT_START
Request	Request a data that is not in subscription list.
Routing	Defines message ID to be rerouted.
Debug	Messages relating to debug features (see DBG system module description in Table 5).

Table 5: Orion UBP Message Types

SOFTWARE PORTABILITY

The Orion architecture supports software porting of non-hardware specific subsystems to other processors with ease. Orion distributed mode, also known as host mode, allows the division of the non-hardware specific subsystems to work in a separate host environment. Separate MA's, which is independent of the application, can exist in both the embedded and host Operating Systems (OS) to manage the messaging tasks of the local subsystem modules.

A Portable RTOS (UPOS) resides in both processors in a distributed system and provides a message and API abstraction layer to the VS_DSP RTOS. This allows location of modules in one OS to be transparent to modules running in the other OS.

All of the software application code will execute on top of the UPOS and use it for RTOS features such as message queues and threading operations. Direct access to the VS_DSPTM RTOS features is not required, which reduces the complexity of migrating and developing software components on different processor environment.

SOFTWARE DEVELOPMENT KIT

A complete software development kit (SDK) is available for the Orion navigation software package. The Orion SDK allows users to add, modify, or remove individual subsystems in order to customize the software for specific applications.

The Orion SDK includes sample source code and development tools to add and initialize logical units, and customize messages between logical units. It includes debug features providing direct read/write memory access on embedded side and execution time (CPU usage) profile information.

The SDK tool set includes the Orion Workbench utility which allows development and debugging of subsystems in Windows® host environment; and an easily modifiable compilation environment (Optimizing C Compiler, profiler, linker, debugger) with u-Nav Vega™ and Microsoft® Visual Studio®.

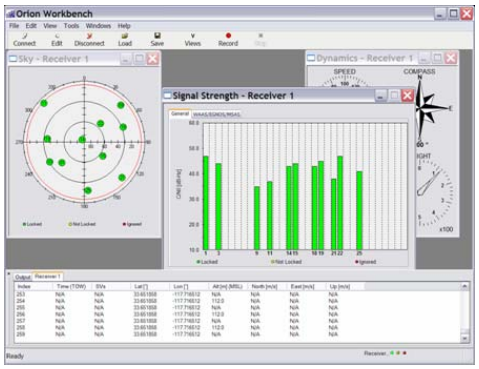


Figure 2: Orion Workbench

FEATURES OF THE ORION SDK

The standard Orion SDK supports the following features and capabilities:

Message Management – The SDK allows users to create and edit communication messages in either the NMEA or UBP format. The User can edit existing NMEA messages and/or create new ones. SDK Users can also set both NMEA and UBP message masks at compile time.

I/O Control – u-Nav GPS chipsets include a variety of user programmable peripherals including general purpose I/O (GPIO), timers, counters, PWM, ADC, and PMU. The SDK allows users to configure and manage these I/O resources as required by a given application.

Subsystem Creation – The SDK allows users to create and add new subsystems to the Orion software.

Flash Memory Support – The SDK allows the User to do a number of modifications related to flash memory devices. In addition to the flash devices supported by the standard Orion software, the SDK allows users to modify the software to support new flash devices. The SDK also allows users to select or customize the size of the flash file system; and read/write application data to the flash device.

Data Editing – The User can edit position, velocity, and time information prior to sending it out (in NMEA subsystem.)

Change the external-RTC driver to support a different external-RTC device

Application Profile – The SDK allows the User to create change configuration of the software to create a custom application profile.

Compiling – The SDK allows the User to compile/build a binary for any supported combination of compile-time options such as frequency plan, RF device type, Flash device type etc.

APPLICATION EXAMPLE 1: SPEED CAMERA DETECTOR

In some countries, speed cameras are distributed throughout cities (common in Korea and Japan) and when a motorist exceeds the speed limit a photograph is taken and a ticket is issued through the mail. The speed cameras are in fixed known locations so it is possible to use GPS to warn a motorist when they are in the vicinity of a speed camera.

Using the SDK, a customer can modify the standard Orion software to specifically support this speed camera detector application. The main areas of modification fall into these categories:

- Read/Write access to non-volatile data storage
- Creating/modifying commands
- Ability to add event-driven algorithms
- User notification of events via peripherals (GPIO, SPI, UART, etc.)

Non-volatile Data Storage – For this application,

the camera location database will be programmed into flash at production time. During the lifetime of the application, elements of that database may need to be deleted, modified and/or appended. The SDK is used to create the subsystem that manages and controls the maintenance of the database in the application flash. In addition, configuration settings of the application may also require modification and storage to flash.

Creating/Modifying Commands - The SDK is also used to extend the NMEA and/or UBP interface to include additional read and write commands. Such commands will include configurations commands sent to the receiver, downloading of large binary or ASCII files to and from the receiver and event notification commands via the UART from the receiver to the end user.

Event-Driven algorithms - The application may use the GPS processor to algorithms to compare the current position with positions in the database. The algorithms run at regular intervals based on a new fix or a timer interrupt. The algorithms will include high speed search of the database and range calculations to the current position.

User Notification - The application will notify the end user when the current position is within a preconfigured range of a stored position. The SDK can be used to program I/O to enable LED's, alarms, synthesized speech, or text messages; as well as sending a message via the SPI port and/or UART.

APPLICATION EXAMPLE 2: POSITION PINNING

In many applications, the position and/or velocity is represented as point on a display. To make that point appear more visually stable and/or accurate to the end user, position pinning is implemented. In position pinning the reported position will not change unless some user-defined criteria are met.

Using the SDK, a customer can modify the standard Orion software to implement position pinning. The main areas of modification fall into these categories:

- Read/Write access to non-volatile data storage
- Creating/modifying commands
- Temporary storage of "fixed" position in near memory
- Position modification
- User Configuration Storage

Non-volatile Data Storage – Once the thresholds for pinning distance and velocity are determined for the application, the SDK is used to add a subsystem that stores these threshold parameters to flash. The subsystem must allocate space in the storage device and then write the parameters to the allocated space.

Creating/Modifying Commands - The SDK is used to add and/or modify commands (NMEA or UBP) to set and query the pinning parameters stored in flash, as well as a pinning enable and disable command.

Position Storage - The modified software will need to save the fixed position information in volatile memory for run-time access.

Position Modification – A task must be developed that will compare the current position against the "fixed" (or first) position. If a distance criterion is exceeded or a velocity threshold is exceeded, the new position is passed through, otherwise the command strings are created with the fixed position and the velocity is set to zero. If the any of the thresholds are exceeded, the "fixed" position is updated to the current position.

SDK PACKAGE

The Orion SDK includes two (2) development kits, two (2) power supplies, two (2) active antennae, SDK User Manual, and the SDK CD with Orion binary and source libraries.

Figure 3: Development Kit

SOFTWARE LICENSE AND SUPPORT

The Orion software and SDK is developed by u-Nav microelectronics and is available under u-Nav license. The software is delivered in binary code format, and partially in sources code for SDK usage. The SDK is supported with training, documentation, software, examples, and reference design hardware.

u-Nav Microelectronics strives to produce quality documentation and welcomes your feedback. Please send comments or suggestions to tech.pubs@unav-micro.com.

For further information, please contact:

Sales:

US.Sales@unav-micro.com
Europe.Sales@unav-micro.com
Korean.Sales@unav-micro.com
Japan.Sales@unav-micro.com
SoutheastASIA.Sales@unav-micro.com
Australia.Sales@unav-micro.com
NewZealand.Sales@unav-micro.com
India.Sales@unav-micro.com
China.Sales@unav-micro.com

Technical Support:

Technicalsupport.asia@unav-micro.com
TechnicalSupport.US@unav-micro.com
TechnicalSupport.Europe@unav-micro.com

Corporate Office:

u-Nav Microelectronics Corporate Headquarters
8 Hughes Irvine, Ca. 92618
Phone: +1 (949) 453-2727

Marketing:

Marketing@unav-mico.com

Website:

http://www.unav-micro.com/

瑞士u-blox公司 全线产品代理

联系方式:

洪 维
King Hong

市场部
15012591515

飞扬科技(香港)有限公司
深圳市蝴蝶谷科技有限公司
Shenzhen Rise Technology co.,LTD
地址: 深圳市福田区车公庙工业区201栋东座7楼
电话: 0755-81306214 传真: 0755-83318188
E-mail: GPSbaby@gmail.com http://www.GPSbaby.com

主要产品线:

U-blox 全系列GPS模块

U-blox GPS原装板卡/评估套件

U-blox GPRS模块

RISE-GPS系列u-blox GPS模块评估板
专业化成品GPS板卡

RISE-M系列GPS模块
大客户订制GPS解决方案

RISE 800MHz高速ARM11嵌入式CPU
支持DDR2代内存及H.264硬解压和2D及3D图形加速

MID及UMPC方案
PCBA方式及专业产品大客户订制模式

GPS电子罗盘成品
GPS导航仪,GPS电子警示器