

# RTKLIB Manual: Demo5 version

This document is based on the RTKLIB 2.4.2 manual by Tomoji Takasu available at:

[http://www.rtklib.com/prog/manual\\_2.4.2.pdf](http://www.rtklib.com/prog/manual_2.4.2.pdf)

It is updated to reflect differences between the RTKLIB 2.4.2 code and the Demo5 code.



Dec 21, 2021

## Contents

1	Overview .....	1
2	User Requirements.....	3
2.1	System Requirements .....	3
2.2	License .....	4
3	Instructions .....	5
3.1	Installation and Uninstallation .....	5
3.2	Real-Time Positioning with RTKNAVI.....	7
3.3	Configure Input, Output and Log Streams for RTKNAVI.....	23
3.5	Configure Positioning Options for RTKNAVI and RTKPOST .....	35
3.6	Convert Receiver Raw Data to RINEX with RTKCONV .....	53
3.7	View and Plot Solutions with RTKPLOT .....	58
3.8	View and Plot Observation Data with RTKPLOT .....	71
3.9	Download GNSS Products and Data with RTKGET .....	79
3.10	NTRIP Browser.....	85
3.11	Use CUI APs of RTKLIB .....	88
4	Build APs or Develop User APs with RTKLIB .....	89
4.1	Rebuild GUI and CUI APs on Windows .....	89
4.2	Build CUI APs.....	90
4.3	Develop and Link User APs with RTKLIB.....	91
	Appendix A CUI Command References .....	92
A.1	RTKRCV .....	92
A.2	RNX2RTKP .....	95
A.3	POS2KML.....	97

A.4	CONVBIN .....	98
A.5	STR2STR .....	101
Appendix B	File Formats .....	103
B.1	Positioning Solution File.....	103
B.2	SBAS Log File.....	106
B.3	Solution Status File.....	108
B.4	Configuration File .....	111
B.5	URL List File for GNSS Data.....	114
Appendix C	API References .....	116
Appendix D	Files and Messages .....	124
D.1	Supported RINEX Files.....	124
D.2	Supported Receiver Messages .....	125
D.3	Supported Signal IDs/Observation Types.....	127
D.4	Default Priorities of Multiple Signals .....	129
D.5	Receiver Dependent Input Options .....	130
Appendix E	Models and Algorithms .....	132
E.1	Time System.....	134
E.2	Coordinates System .....	137
E.3	GNSS Signal Measurement Models .....	140
E.4	GNSS Satellite Ephemerides and Clocks.....	145
E.5	Troposphere and Ionosphere Models.....	151
E.6	Single Point Positioning.....	156
E.7	Kinematic, Static and Moving-Baseline .....	163
Appendix F	Notes on Config Parameters .....	179
Appendix G	GNSS Signal Specifications .....	188
References	.....	190

# 1 Overview

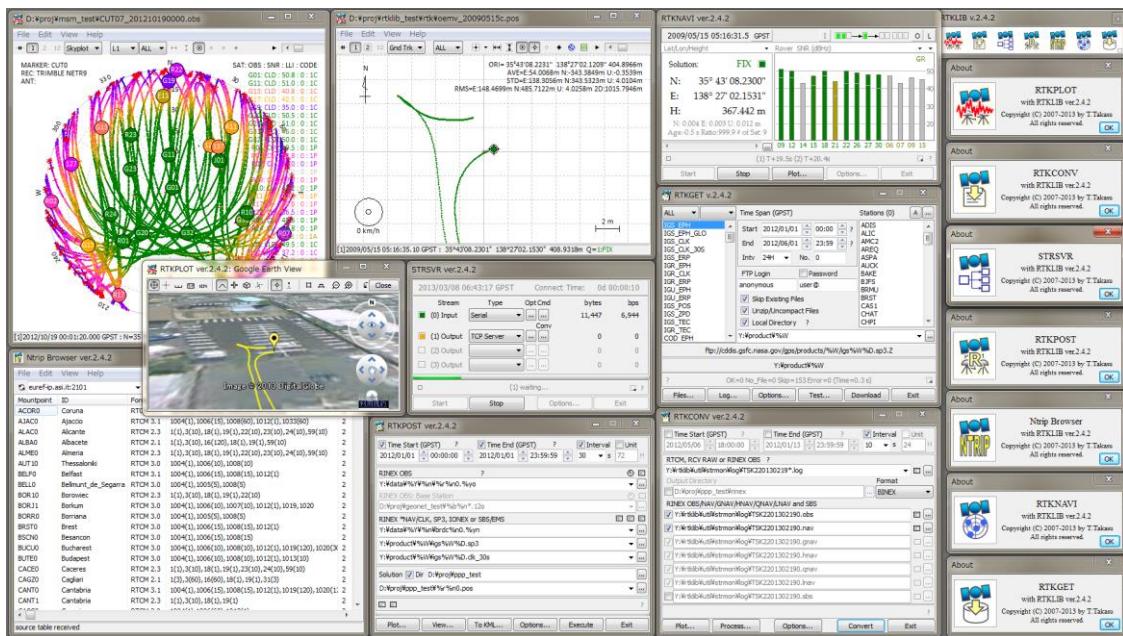
RTKLIB is an open source program package for standard and precise positioning with GNSS (global navigation satellite system). RTKLIB consists of a portable program library and several APs (application programs) utilizing the library. The features of RTKLIB are:

- (1) It supports standard and precise positioning algorithms with:  
GPS [1][2][3], GLONASS [4], Galileo [5], QZSS [6], BeiDou [7] and SBAS [8]
- (2) It supports various positioning modes with GNSS for both real-time- and post-processing:  
Single, DGPS/DGNSS, Kinematic, Static, Moving-Baseline, Fixed, PPP-Kinematic, PPP-Static and PPP-Fixed.
- (3) It supports many standard formats and protocols for GNSS:  
RINEX 2.10 [9], 2.11 [10], 2.12 [11] OBS/NAV/GNAV/HNAV/LNAV/QNAV, RINEX 3.00 [12], 3.01 [13], 3.02 [14] OBS/NAV, RINEX 3.02 CLK [15], RTCM ver.2.3 [16], RTCM ver.3.1 (with amendment 1-5) [17], RTCM ver.3.2 [18], BINEX [19], NTRIP 1.0 [20], NMEA 0183 [21], SP3-c [22], ANTEX 1.4 [23], IONEX 1.0 [24], NGS PCV [25] and EMS 2.0 [26] (refer Appendix D.1 and D.2 for details).
- (4) It supports several GNSS receivers' proprietary messages:  
NovAtel [27]: OEM4/V/6, OEM3, OEMStar, Superstar II, Hemisphere [28]: Eclipse, Crescent, u-blox [29]: LEA-4T/5T/6T, SkyTraq [30]: S1315F, JAVAD [31] GRIL/GREIS, Furuno [32] GW-10-II/III and NVS [33] NV08C BINR (refer Appendix D.2 for details).
- (5) It supports external communication via:  
Serial, TCP/IP, NTRIP, local log file (record and playback) and FTP/HTTP (automatic download).
- (6) It provides many library functions and APIs (application program interfaces):  
Satellite and navigation system functions, matrix and vector functions, time and string functions, coordinates transformation, input and output functions, debug trace functions, platform dependent functions, positioning models, atmosphere models, antenna models, earth tides models, geoid models, datum transformation, RINEX functions, ephemeris and clock functions, precise ephemeris and clock functions, receiver raw data functions, RTCM functions, solution functions, KML converter, SBAS functions, options functions, stream data input and output functions, integer ambiguity resolution, standard positioning, precise positioning, post-processing positioning, stream server functions, RTK server functions, downloader functions.
- (7) It includes the following GUI (graphical user interface) and CUI (command-line user interface) APs. () shows the section describing the instruction for each AP in the manual.

## RTKLIB Demo5 Manual

Function	GUI AP	CUI AP	Notes
(a) AP Launcher	RTKLAUNCH (3.1)	-	
(b) Real-Time Positioning	RTKNAVI (3.2, 3.3, 3.5)	RTKRCV (3.11, A.1)	
(c) Communication Server	STRSVR, (3.3)	STR2STR (3.11, A.5)	
(d) ) Post-Processing Analysis	RTKPOST (3.4, 3.5)	RNX2RTKP (3.11, A.2)	
(e) RINEX Converter	RTKCONV (3.6)	CONVBIN (3.11, A.4)	
(f) Plot Solutions and Observation Data	RTKPLOT (3.7, 3.8)	-	
(g) Downloader for GNSS Products and Data	RTKGET (3.9)	-	
(h) NTRIP Browser	SRCTBLBROWS (3.10)	-	

- (8) All of the executable binary APs for Windows are included in the package as well as whole source programs of the library and the APs.



RTKLIB GUI APs on Windows 7

## 2 User Requirements

### 2.1 System Requirements

The executable binary GUI and CUI APs included in the package require Microsoft Windows [35] environment. On the other OS or environment, you have to compile and build CUI APs by yourself.

All of the library functions and APIs were written in ANSI C (C89). The library internally uses winsock and WIN32 thread for Windows with the compiler option -DWIN32 and the standard socket and pthread (POSIX thread) for Linux/UNIX without any option. By setting the compiler option -DLAPACK or -DMKL, the library uses LAPACK/BLAS [36] or Intel MKL [37] for fast matrix computation.

The CUI APs were also written in ANSI C. The library and CUI APs can be built on many environments like gcc on Linux. The GUI APs were written in C++ and utilize Embarcadero/Borland VCL (visual component library) [38] for GUI toolkits. All of the executable binary APs in the package were built by Embarcadero C++ builder on Windows 10.

The executable GUI APs were tested on Windows 7 (64bit). The CUI APs were also built and tested on Ubuntu [39] 11.04 Linux and x86 CPU.

Notes: Previous versions of RTKLIB until ver. 2.4.1 were built by a free edition of Borland C++ (Turbo C++ 2006). Turbo C++, however, is no longer supported in ver. 2.4.2 because of type incompatibility problem of GUI strings between ver.2.4.2 and the previous ones.

## 2.2 License

The RTKLIB software package is distributed under the following BSD 2-clause license [40] and additional two exclusive clauses. Users are permitted to develop, produce or sell their own non-commercial or commercial products utilizing, linking or including RTKLIB as long as they comply with the license.

Notes: Previous versions of RTKLIB until ver. 2.4.1 had been distributed under GPLv3 [59] license.

---

Copyright (c) 2007-2021, T. Takasu, All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- The software package includes some companion executive binaries or shared libraries necessary to execute APs on Windows. These licenses succeed to the original ones of these software.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 3 Instructions

### 3.1 Installation and Uninstallation

- (1) Extract the program package `rtklib_<ver>.zip` or `rtklib_<ver>_bin.zip` to appropriate directory `<install dir>` (`<ver>` indicates the version number). The RTKLIB directory structure is as follows.

```
rtklib_<ver>
  \src           : Source programs of RTKLIB library *
    \rcv          : Source programs depending on GPS/GNSS receivers *
    \bin          : Executable binary APs and DLLs for Windows
    \data         : Sample data for APs
    \app          : Build environment for APs *
      \rtknavi     : RTKNAVI      (GUI) *
      \rtknavi_mkl : RTKNAVI_MKL (GUI) *
      \strsvr      : STRSVR       (GUI) *
      \rtkpost     : RTKPOST      (GUI) *
      \rtkpost_mkl : RTKPOST_MKL (GUI) *
      \rtkplot      : RTKPLOT       (GUI) *
      \rtkconv      : RTKCONV      (GUI) *
      \srctblbrows : NTRIP Browser (GUI) *
      \rtkget       : RTKGET       (GUI) *
      \rtklaunch   : RTKLAUNCH   (GUI) *
      \rtkrcv      : RTKRCV       (CUI) *
      \rnx2rtkp    : RNX2RTKP    (CUI) *
      \pos2kml     : POS2KML     (CUI) *
      \convbin     : CONVBIN     (CUI) *
      \str2str      : STR2STR      (CUI) *
      \appcmn      : Common routines for GUI APs *
      \icon        : Icon data for GUI APs *
    \lib          : Libraries generation environment*
    \test         : Test program and data *
    \util         : Utilities *
    \doc          : Document files
```

\* Not included in the binary package `rtklib_<ver>_bin.zip`

- (2) Create the shot-cuts of the GUI AP executable binaries in `<install dir>\rtklib_<ver>\bin`. To execute CUI APs, add `<install dir>\rtklib_<ver>\bin` to the command path.
- (3) The GUI and CUI APs in RTKLIB never utilize the Windows registry. To uninstall the package, simply delete the whole files and directories in the installed directory.
- (4) Optional settings for GUI APs are saved in INI files (\*.ini) usually placed in the directory `<install dir>\rtklib_<ver>\bin`. Note that the directory for the INI files is changed in ver. 2.4.2. To succeed your optional settings for the previous version RTKLIB APs, please copy the INI files (\*.ini) in `c:\Windows` to the directory `<install dir>\rtklib_<ver>\bin`.
- (5) Some GUI APs (RTKCONV, RTKPOST, RTKNAVI, RTKGET and STRSVR) can be executed with the command line option `-i <inifile>.ini` to select an alternative INI file for another optional settings. You can switch options for such APs by using the command line option as well as the option `-t <title>` to change the window title. So you can configure multiple short-cuts for the same GUI AP with different options by setting the properties of these short-cuts.
- (6) To use RTKPOST\_MKL or RTKNAVI\_MKL, which is the Intel-MLK-library-linked version RTKPOST or RTKNAVI for faster matrix computation on multiple-core CPU or multiple-processor PCs, please set the Windows environment variable `OMP_NUM_THREADS` to **2**, **4** or **8** according to the number of the CPU cores. That enables multi-threading matrix computation in order to shorten processing time.
- (7) To execute GUI APs easily, an AP launcher application RTKLAUNCH is added in ver. 2.4.2. To run RTKLAUNCH, execute `<install dir>\rtklib_<ver>\bin\rtklaunch.exe`. You can click AP icons in the RTKLAUNCH window or select pop-up menus at the task-tray-icon in order to execute RTKLIB APs. RTKLAUNCH accepts the option `-mkl` to launch RTKPOST\_MKL and RTKNAVI\_MKL instead of RTKPOST and RTKNAVI, and the option `-tray` to start the launcher as a task-tray-icon.

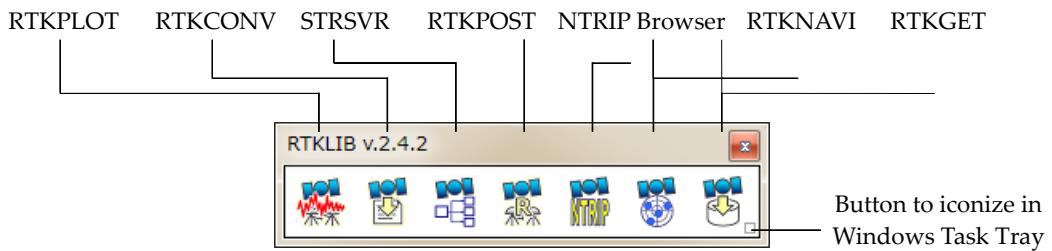


Figure 3.1-1 RTKLAUNCH window and launcher icons for APs

## 3.2 Real-Time Positioning with RTKNAVI

An real-time positioning AP RTKNAVI inputs raw observation data of GPS/GNSS receivers and execute navigation processing in real-time. By setting the positioning mode to Kinematic and configuring the rover and the base station receiver data inputs, RTK-GPS/GNSS is enabled with OTF (on-the-fly) integer ambiguity resolution.

- (1) Execute the binary AP file `<install dir>\rtklib_<ver>\bin\rtknavi.exe`. (double-click the icon or full in the path in the command console) You can see the main window of RTKNAVI. You can also execute `<install dir>\rtklib_<ver>\bin\rtknavi_mkl.exe` instead.

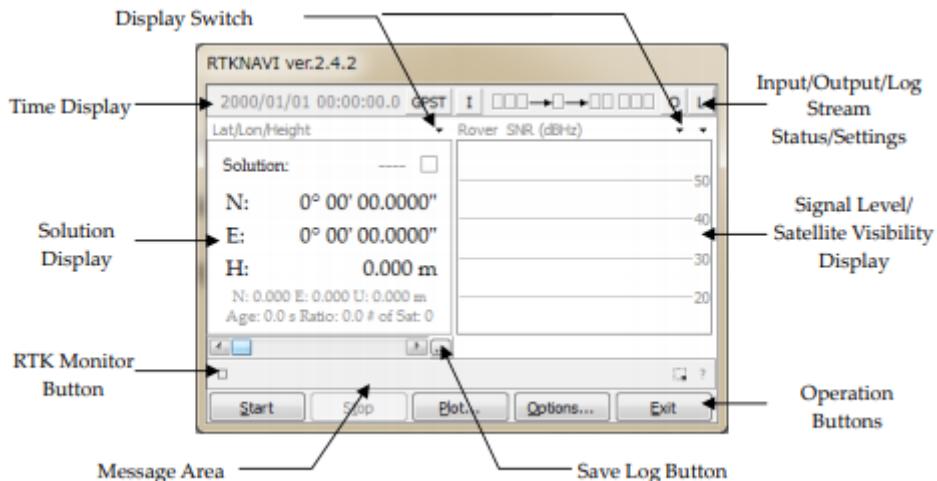


Figure 3.2-1 Main Window of RTKNAVI

- (2) The following figure shows the data flow of RTKNAVI. You have to set up Input Streams, Output Streams (optional) and Log Streams (optional) for real-time positioning. Refer 3.3 Configure Input, Output and Log Streams for RTKNAVI for several sample configurations of these streams.

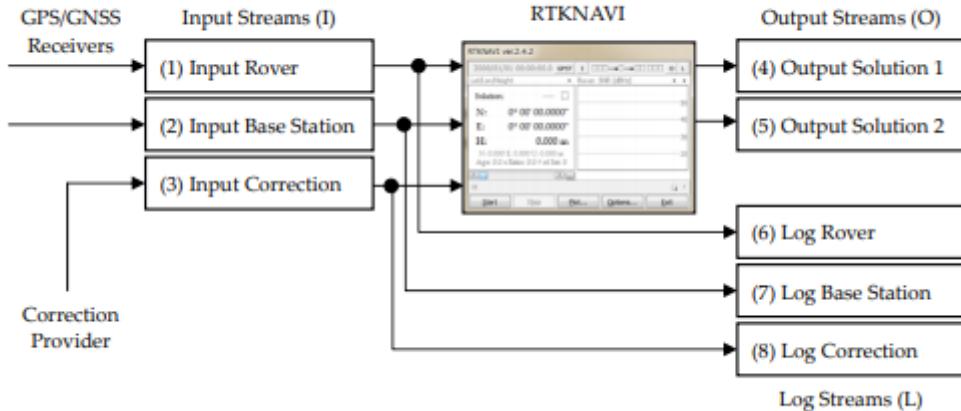


Figure 3.2-2 Data Flow of RTKNAVI

- (3) For real-time positioning with RTKNAVI, you have to input the raw observation data and satellite ephemerides from the GPS/GNSS receivers. To set the input stream, push the button **I** upper center in the main window. You can see the "Input Streams" dialog.

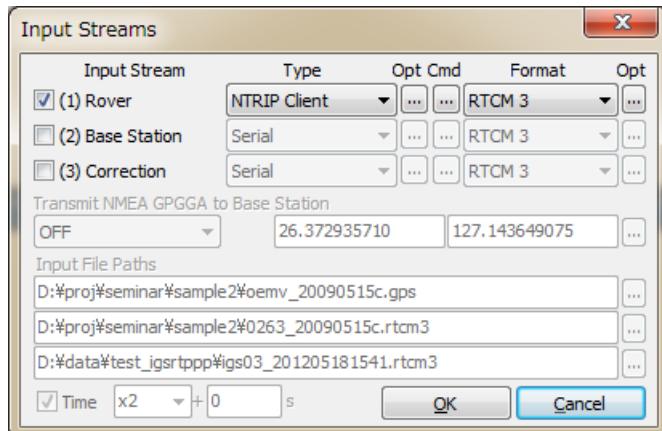


Figure 3.2-3 Input Streams Dialog of RTKNAVI

- (4) Check and set the stream type of Rover, Base-station or Correction in the dialog. If you set the "Positioning Mode" option "Single", the input streams for "Base-station" and "Correction" are not required. The stream types can be selected from the following options.

- (a) Serial : Input data from a serial port (RS232C or USB)
- (b) TCP Client : Connect to a TCP server and input data via the TCP connection
- (c) TCP Server : Accept a TCP client connection and input data via the TCP connection
- (d) NTRIP Client : Connect to a NTRIP caster [20] and input data via the NTRIP.

NRTK (network RTK) server supporting NTRIP and RTCM 2/3 can also be used for the base-station via Internet.

- (e) File : Input data from a log file.
- (f) FTP : Input data after downloading a file by FTP (Only for Correction)
- (g) HTTP : Input data after downloading a file by HTTP (Only for Correction)

You have to select the stream data format from the following options with the pull down menu under "Format". Refer Appendix D.2 for supported messages by RTKLIB. You shall configure your GPS/GNSS receivers to output at least GPS/GNSS observation data and navigation data (ephemerides). For detailed operation for the receiver settings, refer the appropriate manuals for the GPS/GNSS receivers.

- (a) RTCM2 : RTCM 2.3
- (b) RTCM3 : RTCM 3.0, 3.1 (with amendment 1-5) and 3.2
- (c) NovAtel OEM6 : NovAtel OEM4/V6 and OEMStar binary format
- (e) u-blox : u-blox LEA-4T, 5T and 6T binary format
- (f) Swiftnav : Swiftnav Piksi Multi
- (g) Hemisphere : Hemisphere Crescent/Eclipse binary format
- (h) SkyTraq : SkyTraq S1315F binary format
- (j) Javad : JAVAD GRIL/GREIS binary format
- (k) NVS BINR : NVS NV08C BINR format
- (l) BINEX : BINEX format (only supports big-endian, forward, regular CRC)
- (m) SP3 : SP3 precise ephemeris (only for Correction)

- (5) If you select "Serial" as the stream type, push  button under "Opt" label to set the options of "Port" selection, "Bit-rate", "Byte size", "Parity", number of "Stop bits" and "Flow Control" with the "Serial Options" dialog.

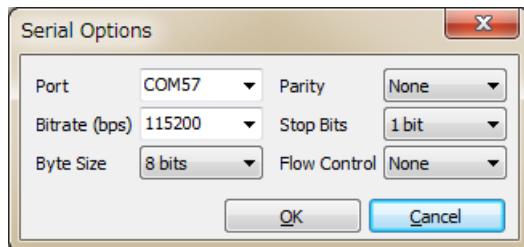


Figure 3.2-4 Serial Option Dialog of RTKNAVI

- (6) In case of selecting "Serial", "TCP Client" or "TCP Server" as the stream type, you can configure the

startup and shutdown commands to be sent to the GPS/GNSS receiver through the stream. To set up the commands, push  button under the "Cmd" label. Fill in commands in the text fields in the "Serial/TCP Commands" dialog. If you do not check "Commands at startup" or "Commands at shutdown", the startup or shutdown command is not sent to the receiver. You can also load the commands from a command file by pushing  button or save the commands to a command file with  button. A command file is just a text file including startup commands and shutdown commands separated by a line starting with "@". Sample command files for some typical GPS/GNSS receivers are found at <install dir>\rtklib\_<ver>/data/\*.cmd.

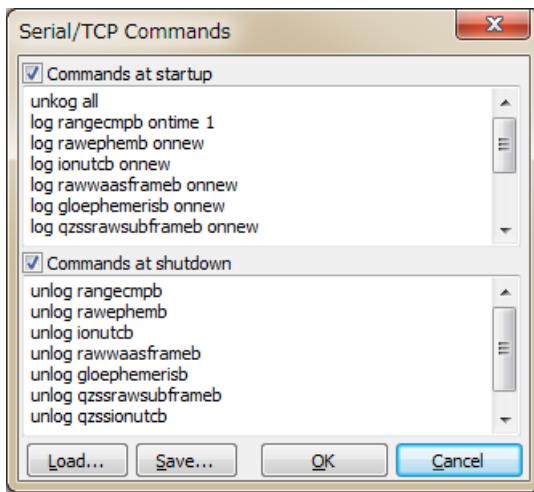


Figure 3.2-5 Serial/TCP Commands Dialog of RTKNAVI

- (7) A line starting with "!" in the commands is treated as a receiver binary command. The following commands can be used for u-blox, SkyTraq and NVS receivers. Refer the receivers' manuals for details.

```
!UBX . . . : u-blox LEA-4T/5T/6T command
!UBX CFG-PRT portid res0 res1 mode baudrate inmask outmask flags
!UBX CFG-USB vendid prodid res1 res2 power flags vstr pstr serino
!UBX CFG-MSG msgid rate0 rate1 rate2 rate3
!UBX CFG-NMEA filter version numsv flags
!UBX CFG-RATE meas nav time
!UBX CFG-CFG clear_mask save_mask load_mask
!UBX CFG-TP interval length status time_ref res adelay rdelay udelay
!UBX CFG-NAV2 ...
!UBX CFG-DAT maja flat dx dy dz rotx roty rotz scale
!UBX CFG-INF protocolid res0 res1 mask0 mask1 mask2 mask3
!UBX CFG-RST navbbr reset res
```

```

!UBX CFG-RXM gpsmode lpmode
!UBX CFG-ANT flags pins
!UBX CFG-FXN flags treacq tacq treacqoff tacqoff ton toff res basetow
!UBX CFG-SBAS mode usage maxsbas res scanmode
!UBX CFG-LIC key0 key1 key2 key3 key4 key5
!UBX CFG-TM intid rate flags
!UBX CFG-TM2 ch res0 res1 rate flags
!UBX CFG-TMODE tmode posx posy posz posvar svinmindur svinvarlimit
!UBX CFG-EKF ...
!STQ ... : SkyTraq S1315F binary command
!STQ RESTART [arg...] system restart
!STQ CFG-SERI [arg...] configure serial port property
!STQ CFG-FMT [arg...] configure output message format
!STQ CFG-RATE [arg...] configure binary measurement output rates
!NVS ... : NVS NV08C binary command
!NVS CFG-PVTRATE [arg...] configure PVT rate
!NVS CFG-RAWRATE [arg...] configure raw data rate
!NVS CFG-SMOOTH configure smooth range
!NVS CFG-BINR xx [...] send binary command for NVS
                           (input hexadecimal series)
!WAIT time : wait for time (ms)

```

- (8) If you select "TCP Client" or "TCP Server" as the stream type, you can set the options of "TCP server address" (for TCP Client only) and the "Port" number with the "TCP Client Options" or "TCP Server Options" dialog. If you select "TCP Server" as the stream type, multiple TCP client connections are allowed.

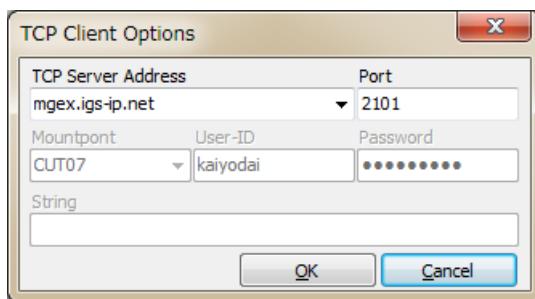


Figure 3.2-6 TCP Client Options Dialog of RTKNAVI

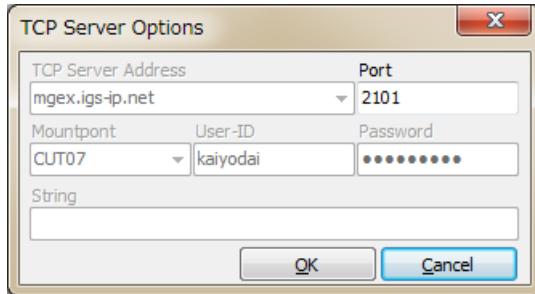


Figure 3.2-7 TCP Server Options Dialog of RTKNAVI

- (9) If you select "NTRIP Client" as the stream type, you can set the options of "NTRIP Caster Host" address, "Port" number, "Mount-point" of NTRIP caster, "User-ID" and "Password" with the "NTRIP Client Options" dialog. If you keep the "Port" field blank, the default port number (2101) of NTRIP is used. Note that "User-ID" cannot contain ":".

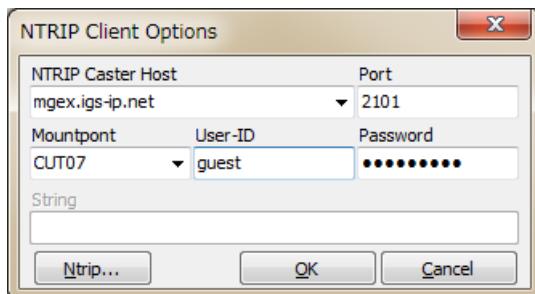


Figure 3.2-8 NTRIP Client Options Dialog of RTKNAVI

- (10) If you select "File" as the stream type, input the file path to the text field Input File Paths. Fill in the path directly or select a file with the file selection dialog by pushing [...] button. The input file should be a receiver raw data log. You can set the replay speed and the start time offset of the log file in Time field (To use the feature, you have to record the log with the time-tag file.)

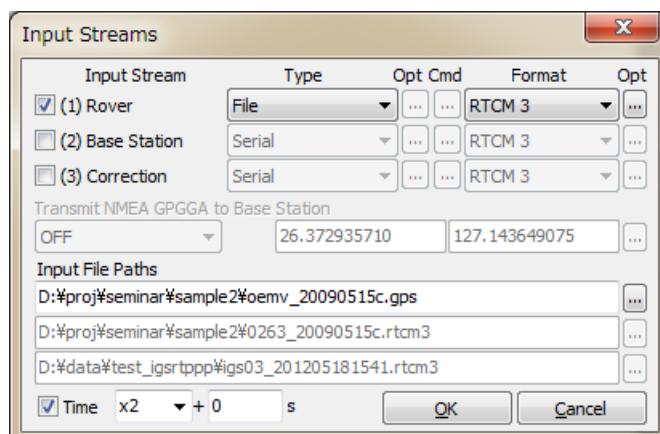


Figure 3.2-9 Input Streams Dialog of RTKNAVI

- (11) By pushing the "Opt" button right of the input stream "Format", you can set receiver-dependent options like "**-GL1X -RL1C -EPHALL**" with the "Receiver Option" dialog. Multiple options can be used separated by spaces. For detailed receiver-dependent options, refer Appendix D.4 Receiver Dependent Input Options. If the input observation data stream contain multiple signals in a frequency, a signal in use for solutions is selected by the default signal priorities without such options. Refer Appendix D.3 Default Priorities for Multiple Signals.

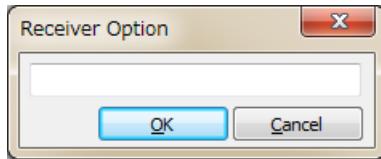


Figure 3.2-10 Receiver Option Dialog of RTKNAVI

- (12) In case of using a NRTK (network RTK) service, which requires NMEA GPGGA messages to select reference station or to setup the VRS (virtual reference station) position, select the message content with the pull down menu at "Transmit NMEA GPGGA to Base Station". If you select "Latitude/Longitude" to send a fixed position, fill in the latitude and longitude of the position for NMEA GPGGA messages in degree (minus means south or west).
- (13) For the correction stream, you can select "FTP" or "HTTP" as the stream type. After pushing "Opt" button, you have to configure FTP or HTTP options with the "FTP Option" or "HTTP Option" dialog. At first, fill in the server address and the file path in the "Download Address" field as the format **<server address>/<file path>**. Usually you might need to include day or time keywords in **<file path>**. For example, in case of downloading IGS ultra-rapid ephemeris from the NASA GSFC CDDIS data server , you can input the download address like:

```
cddis.gsfc.nasa.gov/gps/products/%W/igu%W%D_%hb.sp3.z
```

In this case, the keywords **%W**, **%D** and **%hb** are replaced by GPS week number, day of week and 6 hour of the day according to the download time in GPS Time, respectively. For other keywords which can be used in the file path, push **[?]** button and see the dialog. You can also set Download Interval, Download Offset (for example, Interval = 6H and Offset = 2 H means the download will be tried at 2:00, 8:00, 14:00 and 20:00 in GPS Time), Time Offset in Path for replacing the keywords in the file path, Retry Interval, User (for FTP only) and Password (for FTP only) for the server. For User and Password, "**anonymous**" and your mail address are usually used for anonymous FTP servers.

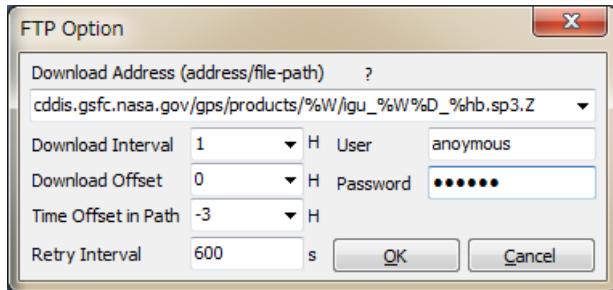


Figure 3.2-11 FTP Option Dialog of RTKNAVI

To use download files, you also have to set the file format in the "Input" dialog. Current version only supports SP3 precise ephemeris for this purpose. Downloaded files are saved in a local directory. The local directory path shall be set with the "Options" dialog - "Files" - "FTP/HTTP Local Directory".

(14) To output of the positioning solutions by RTKNAVI, you shall set the output streams. To set the output streams, push the button upper right in the main window. You can see the "Output Streams" dialog. Check and set the stream type of solution in the dialog. You can configure two independent output streams as maximum. You can select the stream type out of "Serial", "TCP Client", "TCP Server", "NTRIP Server" and "File". The options are similar to the input streams. You also have to select the following output format options. The time and latitude/longitude formats, the type of height and the geoid model and the NMEA interval in output messages also can be configured by the positioning options described in 3.5.

- (a) Lat/Lon/Height : Latitude, longitude and height
- (b) X/Y/Z-ECEF : X/Y/Z components in ECEF frame
- (c) E/N/U-Baseline : E/N/U components of the baseline
- (d) NMEA0183 : NMEA0183 GPRMC, GPGGA, GPGSA, GLGSA, GAGSA, GPGSV, GLGSV and GAGSV

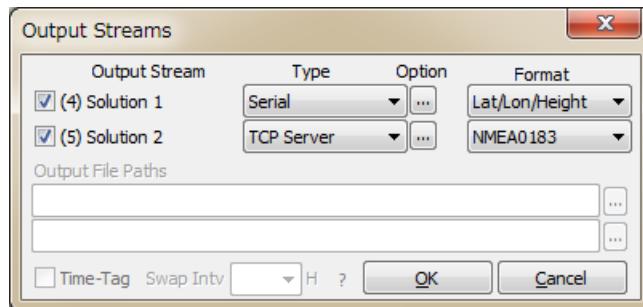


Figure 3.2-12 Output Streams Dialog of RTKNAVI

- (15) If you select "File" as the output stream type, you can include some keywords in the file path to be replaced by date or time. Push **[?]** button to show the keyword replacement in the file paths. If you set the "Swap Intv" option, the output file is swapped periodically in the specified cycle. To use the file swap feature, the file path must contain the keywords to be replaced by the swap time in order to avoid overwriting the previous file.



Figure 3.2-13 Keyword Replacement Dialog of RTKNAVI

- (16) If you select "NTRIP Server" as the output stream type, you have to set NTRIP server options with the "NTRIP Server Option" dialog. The options include the address of "NTRIP Caster Host", the "Port" number to connect NTRIP caster, NTRIP "Mount-point", "Password" and "String" showing the source-table parameters to NTRIP Caster. If the "Port" field blank, the default port number (80) is used. The button **[Ntrip...]** launches the NTRIP browser AP to show the source table provided by the NTRIP caster. Refer 3.10 NTRIP Browser for details.

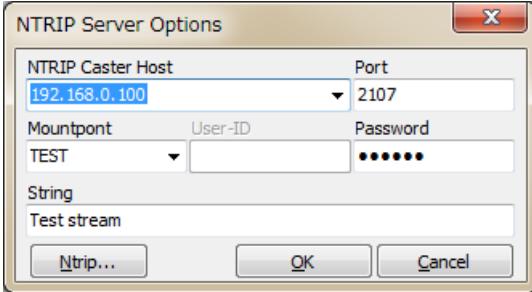


Figure 3.2-14 NTRIP Server Options Dialog of RTKNAVI

- (17) To output an input stream as a path-through log, set the log streams. To configure the log streams, push the button **L** upper right in the main window. You can see the "Log Streams" dialog. The settings are similar to the output streams. If you want to replay the log file as an input stream later, you have to check the "Time-Tag" option and output the time tag file simultaneously. The output path of the time tag file is automatically set to **<output file path>.tag**. The keyword replacements in the file paths and the swap interval are the same as the "Output Streams" dialog.

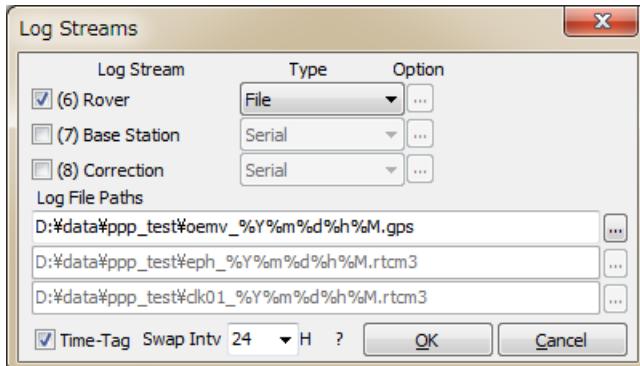


Figure 3.2-15 Log Streams Dialog of RTKNAVI

- (18) To configure the positioning options, push **Options...** button and set the options in the "Options" dialog. For details of the positioning options, refer 3.5 Configure Positioning Options for RTKNAVI and RTKPOST.

- (19) Push **Start** button. The status of each streams are shown on the upper right indicators. From the left, they show the stream/processing status of Input Rover, Input Base Station, Input Correction, the positioning process, Output Solution 1, Output Solution 2, Log Rover, Log Base Station and Log Correction. **Gray** represents not used, **Orange** means waiting for the connection, **Deep-green** means connected or running, **Light-green** means data active (input, output or processing) and **Red** means a

communication error occurs. Some status messages are also shown in the lower center message area in the main window. To stop the positioning process in RTKNAVI push **Stop** button.

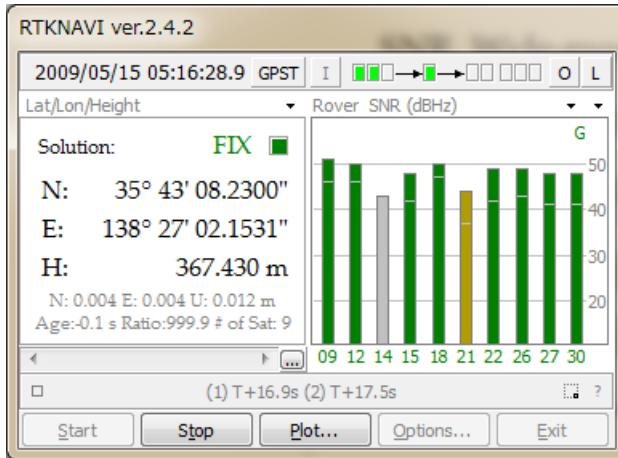


Figure 3.2-16 Main Window of RTKNAVI (RUNNING)

- (20) After the input observation data and ephemerides are completed and valid, RTKNAVI computes the positioning solution and display it in the solution display left in the main window with the solution status (**FIX**, **FLOAT**, **DGPS**, **SBAS**, **SINGLE** or **PPP**), E/N/U or X/Y/Z components of the standard deviation, Age (age of differential), Ratio (ratio factor of ambiguity validation) and # of Sat (number of valid satellites). To switch the format in the solution display, push **▼** button upper right corner. You can switch the solution to Lat/Lon/Height (degree/minute/second), Lat/Lon/Height (degree), X/Y/Z-ECEF (m), E/N/U-Baseline (m), Pitch/Yaw/Length-Baseline (deg, m), alternatively. In the Lat/Lon/Height modes, the labels "H" and "He" indicates the geodetic and ellipsoidal height, respectively.

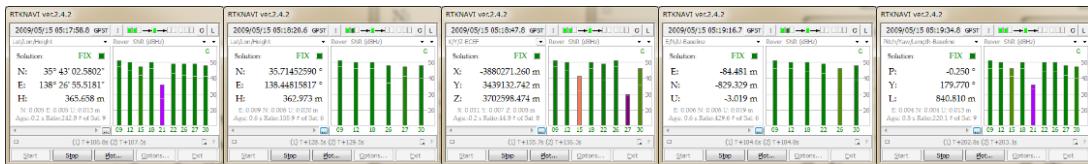


Figure 3.2-17 Solution Status Display of RTKNAVI

- (21) To switch the format in the time display, push **GPST** button upper center in the main window. You can switch the time system to GPST, UTC, LT (local time) and GPST (GPS week/TOW), alternatively.
- (22) In the status display right in the main window, observation SNR (signal to noise ratio, C/N0) status or visible satellites in skyplot are shown. By pushing **▼** button upper right corner of the status display,

## RTKLIB Demo5 Manual

you can switch the contents to Rover : Base SNR, Rover SNR, Rover Skyplot, Base Skyplot and Baseline plot, Wide Rover : Base SNR, Wide-mode Rover SNR, alternatively. You can switch the frequency show in the plot by pushing the second  button near the upper right corner. In the SNR plots or the Skyplots, the colors except for Gray as not-in-use indicate the signal SNR as: > 45 ... 40 ... 35 ... 25 ... < 25 (dBHz). The colors of the satellite IDs in the SNR plots also indicate the satellite system as: **GPS (G)**, **GLONASS (R)**, **Galileo (E)**, **QZSS (J)**, **BeiDou (C)** and **SBAS (S)**.

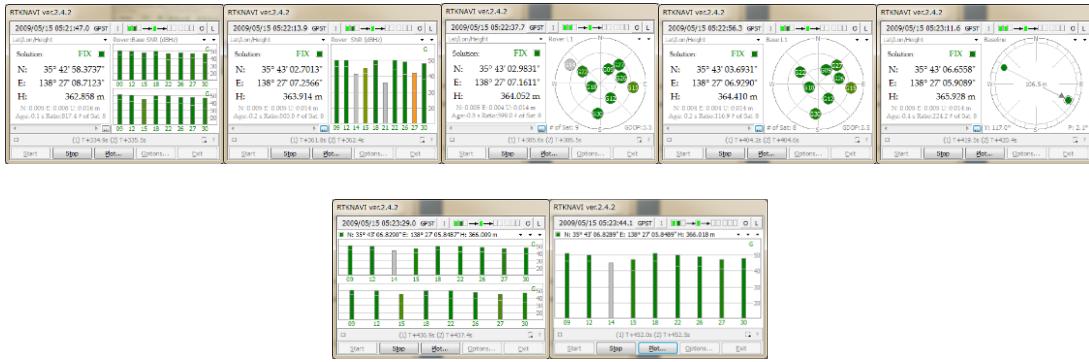


Figure 3.2-18 Satellite and Signal Status Display of RTKNAVI

(23) The size of RTKNAVI window can be changed by dragging the edge of the window. The separator between the left and the right sub-windows can be also dragged to change the layout of the window. To show the status of many of satellites, use this feature introduced in ver. 2.4.2. The font in the Solution sub-window can be changed by the settings by the options dialog the "Options" - "Misc" - "Solution Font".

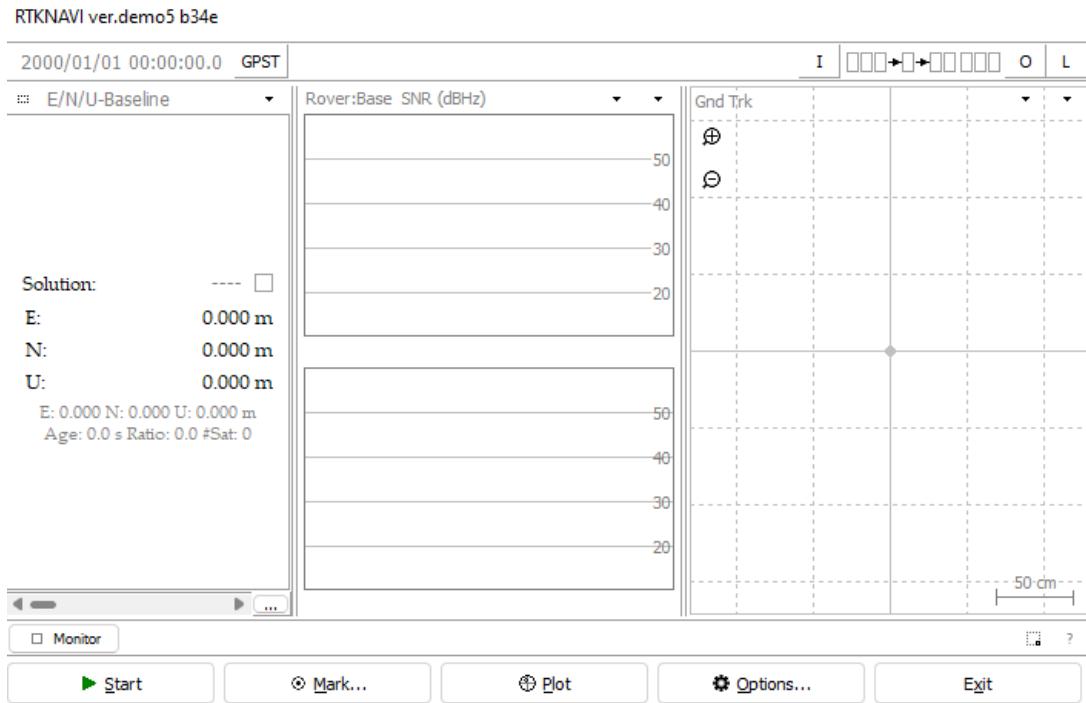


Figure 3.2-19 Main Window of RTKNAVI

- (24) By pushing **[Plot...]** button, you can execute RTKPLOT to plot the current position of the rover receiver on the graph as the real-time solution mode. For details to use RTKPLOT, please refer 3.7 View and Plot Solutions and Observation Data with RTKPLOT.

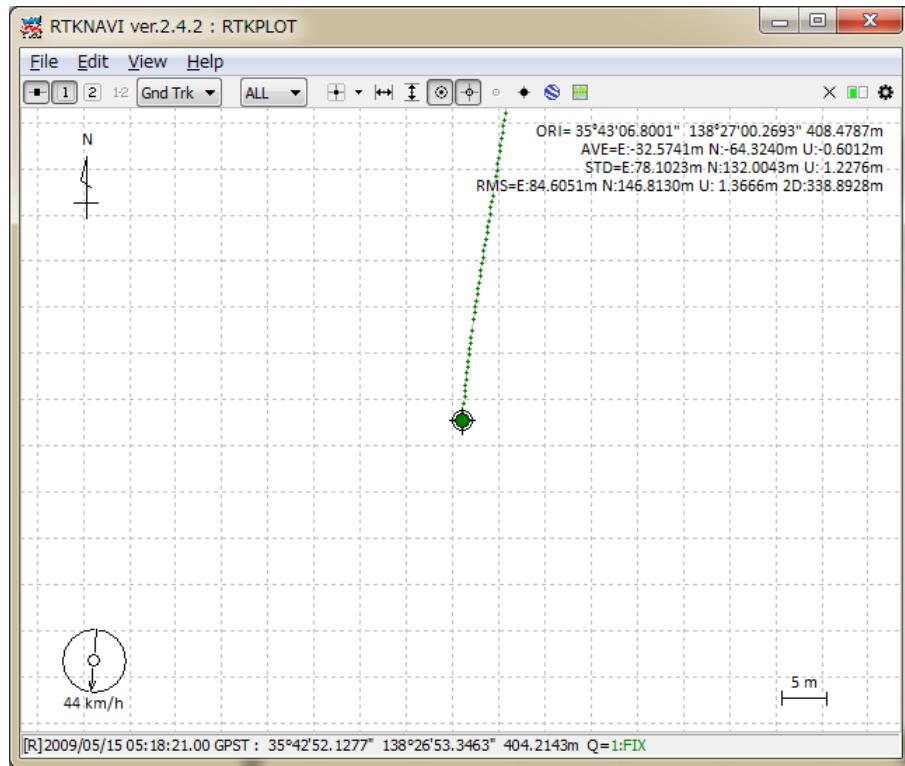


Figure 3.2-20 RTKPLOT Window executed by RTKNAVI

- (25) The positioning solutions are recorded in the internal buffer simultaneously. You can save the internal solution buffer to the file by pushing  below the solution display. The size of the solution buffer and the saved solution log can be configured with the "Options" dialog.
- (26) By pushing the " Monitor" button lower left in the main window, you can see the "RTK Monitor" window. With the window, you can see the internal status of RTKNAVI. You can select the following contents with the upper left pull down menu. Multiple "RTK Monitor" windows are allowed to be shown at the same time. To close the window, push  button.

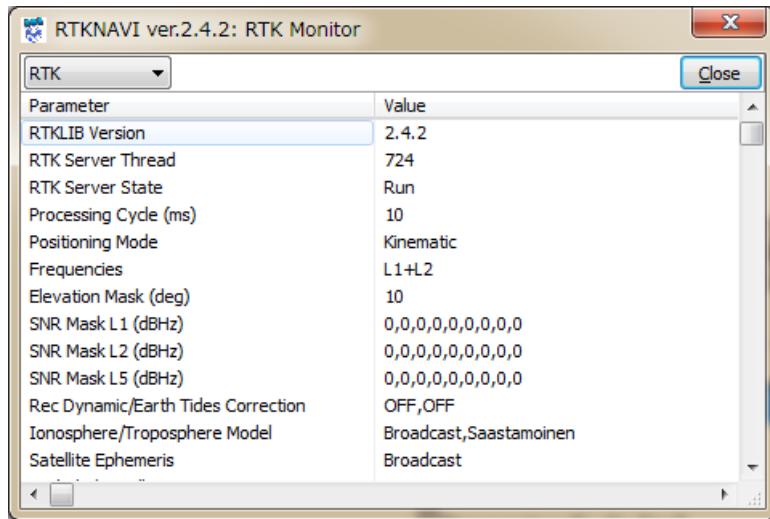


Figure 3.2-21 RTK Monitor Window of RTKNAVI

- (a) RTK : General status of the internal positioning process
- (b) Obs Data : Input observation data. RCV=1 means rover and 2 means base-station
- (c) Nav GPS : GPS satellite navigation messages
- (d) Nav GLONASS : GLONASS satellite navigation messages
- (e) Nav Galileo : Galileo satellite navigation messages
- (f) Nav QZSS : QZSS satellite navigation messages
- (g) Nav BeiDou : BeiDou satellite navigation messages
- (h) Nav GEO : GEO/SBAS satellite navigation messages
- (i) Time/Iono : Time and Ionosphere parameters
- (j) Streams : Status of input, output and log streams
- (k) Sat GPS : Status of GPS satellites
- (l) Sat GLONASS : Status of GLONASS satellites
- (m) Sat Galileo : Status of Galileo satellites
- (n) Sat QZSS : Status of QZSS satellites
- (o) Sat BeiDou : Status of BeiDou satellites
- (p) Sat GEO : Status of GEO/SBAS satellites
- (q) States : State vector values of the estimation filter
- (r) Covariance : Covariance matrix of the estimation filter
- (s) SBAS Msgs : HEX dump of input SBAS messages
- (t) SBAS Long : SBAS long term satellite corrections
- (u) SBAS Iono : SBAS ionospheric delay corrections
- (v) SBAS Fast : SBAS fast corrections
- (w) RTCM Msgs : Status of RTCM 2 or 3 messages

- (x) RTCM DGPS : RTCM DGPS corrections
- (y) RTCM SSR : RTCM SSR corrections
- (z) LEX Msgs : (reserved)
- (aa) LEX Eph/Clock : (reserved)
- (ab) LEX Iono : (reserved)
- (ac) Iono Correction : Ionosphere corrections
- (ad) (1) Rover : Dump of Input Rover stream
- (ae) (2) Base Station : Dump of Input Base Station stream
- (af) (3) Correction : Dump of Input Correction stream
- (ag) (4)(5)Solution 1/2 : Dump of Output Solution 1/2 stream
- (ah) Error/Warning : Error or warning messages

(27) In case of selecting (1) Rover, (2) Base Station or (3) Correction for "RTK Monitor", you can select the input message format with the pull-down menu upper center of the window and dump the messages in the input stream. A dump line shows a message containing the fields: message type, (message length in bytes) : message contents depending on the message format.

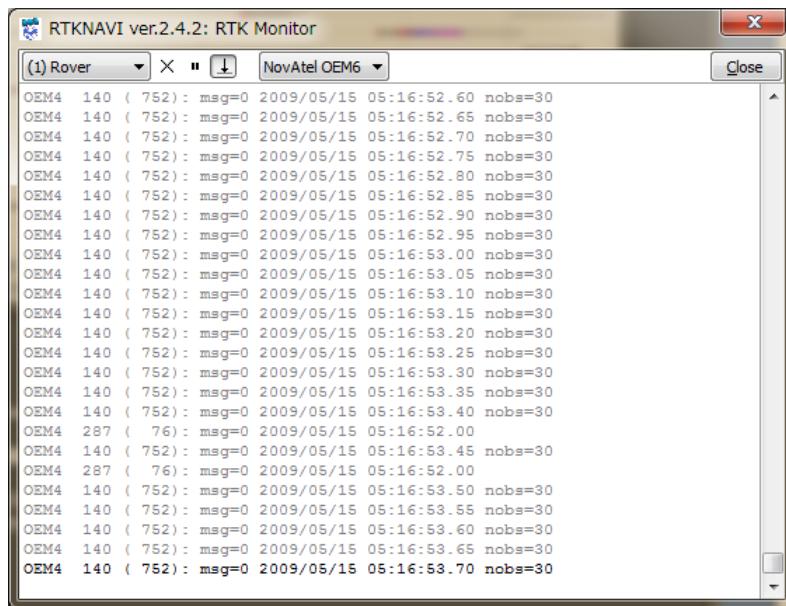


Figure 3.2-22 Stream Dump in RTK Monitor Window of RTKNAVI

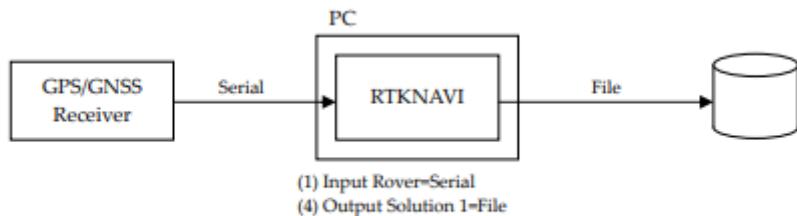
(28) By pushing button lower right in the main window, you can minimize the main window as an icon in the task-tray of Windows desktop. To restore the main window, double-click the task-tray-icon or click right-button on the task-tray-icon and select menu "Main Window...".

### 3.3 Configure Input, Output and Log Streams for RTKNAVI

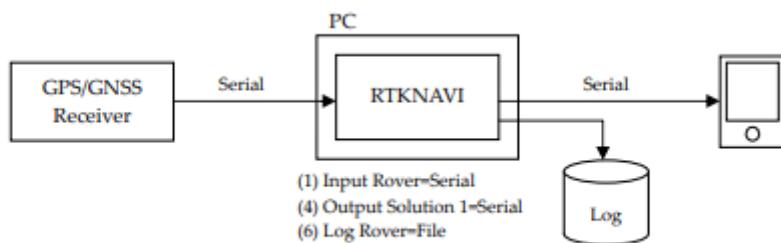
For relative positioning like RTK-GPS/GNSS, the rover receiver and the base station receiver are usually placed on the separated sites. In other cases, user might use the positioning result at the remote site far from these receivers. To interconnect these sites, user has to establish data communication links. To facilitate easy setup of these communication links, RTKLIB provides a communication server utility AP STRSVR, with which user can configure input and output data stream via these communication links. STRSVR also has the function of relay or split the data stream for real-time positioning with RTKNAVI.

For example, to receive observation data of a remote base station at a rover receiver site and to get RTK-GPS solution, user can place a remote PC installing STRSVR connecting to the base station receiver and can configure STRSVR to send data to the rover site. The following examples show the typical applications of RTKNAVI and SVRSVR.

- (1) Single-point positioning and output solutions to a file

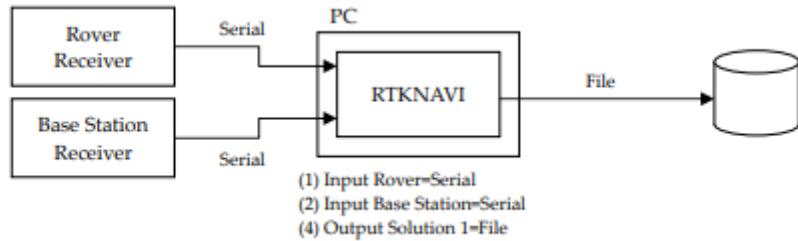


- (2) Single point positioning, output solutions to a serial device, log data to a file

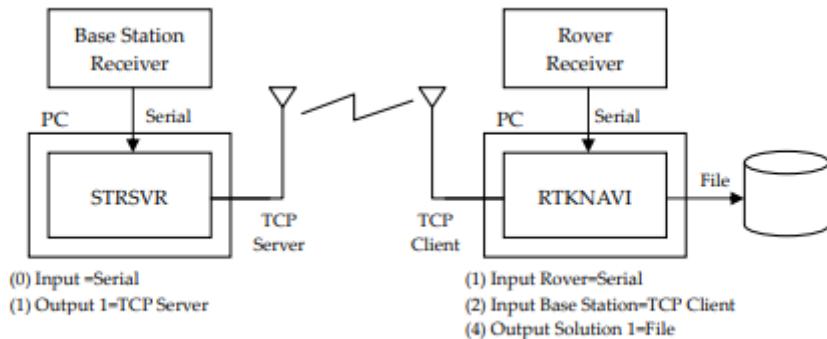


## RTKLIB Demo5 Manual

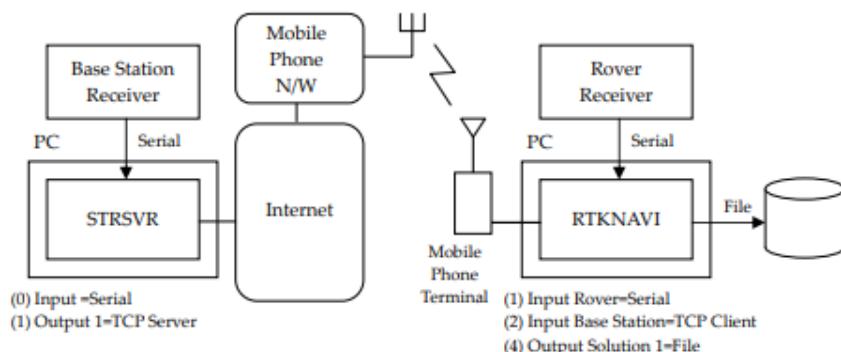
- (3) RTK-GPS/GNSS, input the rover and base-station data from two serial devices



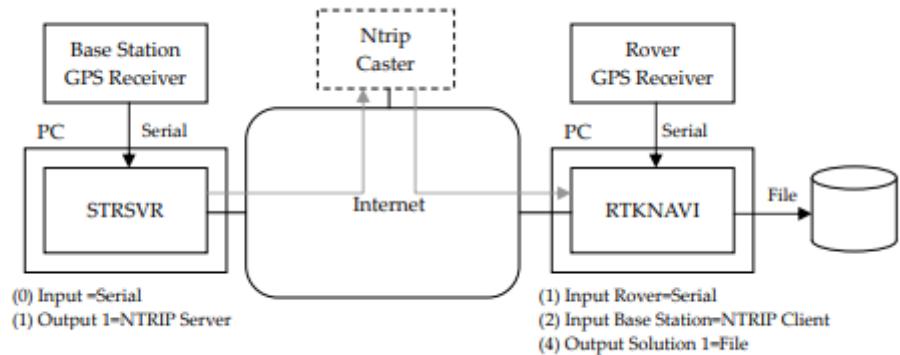
- (4) RTK-GPS/GNSS, input rover data from a serial port and input base-station data from a remote receiver via Wi-Fi network.



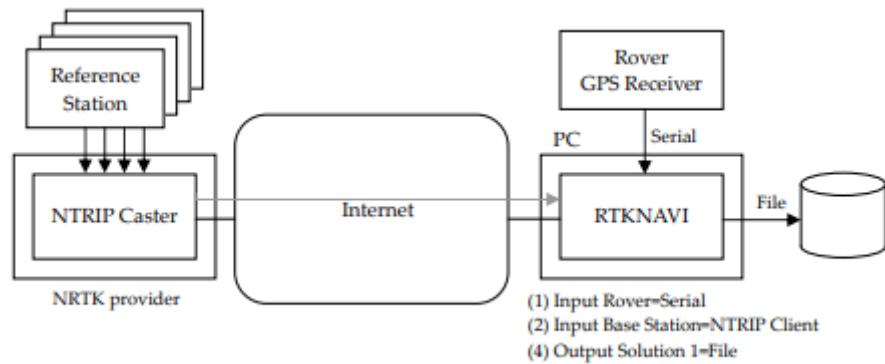
- (5) RTK-GPS/GNSS, input rover data from a serial port and input base station data via mobile phone Internet connection



- (6) RTK-GPS/GNSS, input data from a serial port and input base station data via a NTRIP caster on Internet.

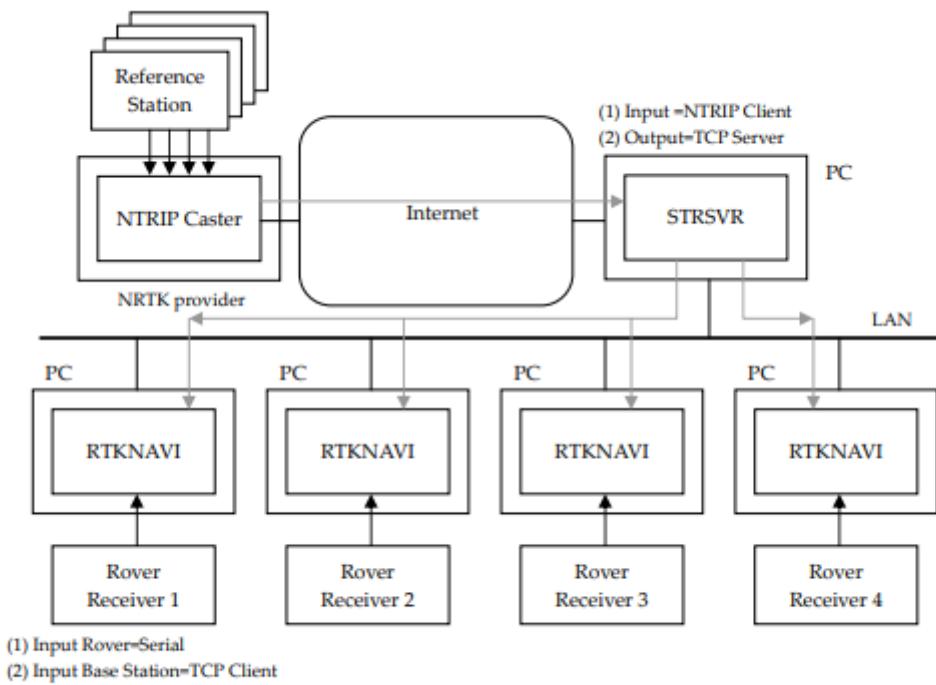


## (7) RTK-GPS/GNSS with NRTK (Network RTK) service via Internet

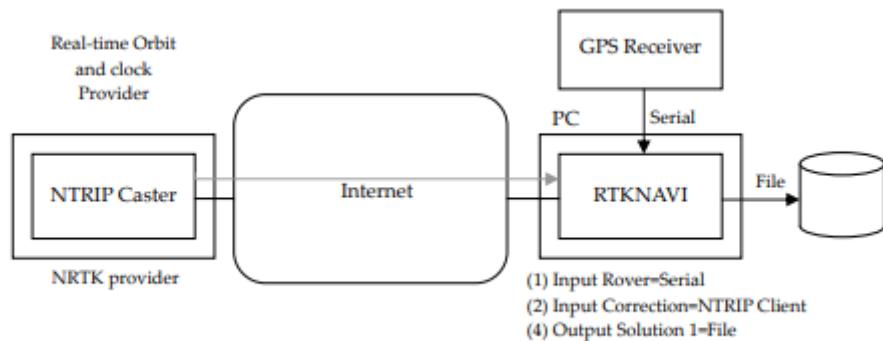


## 8) Multiple RTK-GPS/GNSS with single NRTK service

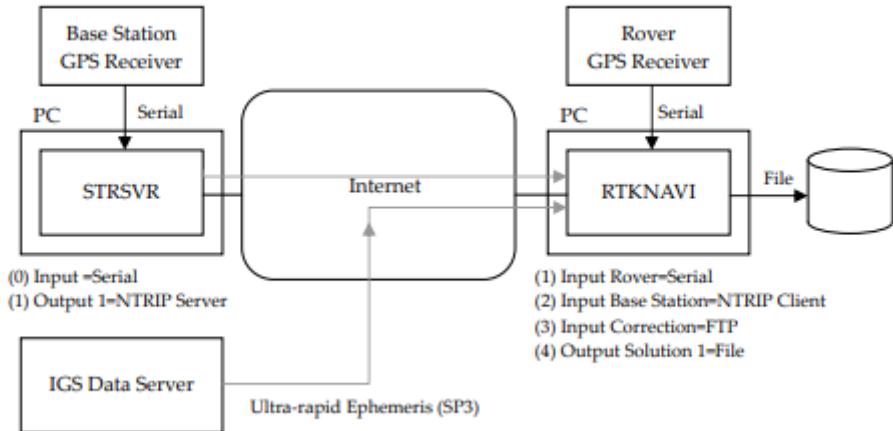
## RTKLIB Demo5 Manual



(9) Real-time PPP with real-time satellite orbit and clock provided as a NTRIP stream.



(10) Long-baseline RTK with FTP download of precise ephemeris



The following instructions are for the operation of STRSVR.

- (1) Execute the binary AP file <install dir>\rtklib\_<ver>\bin\strsvr.exe. You can see the main window of STRSVR.

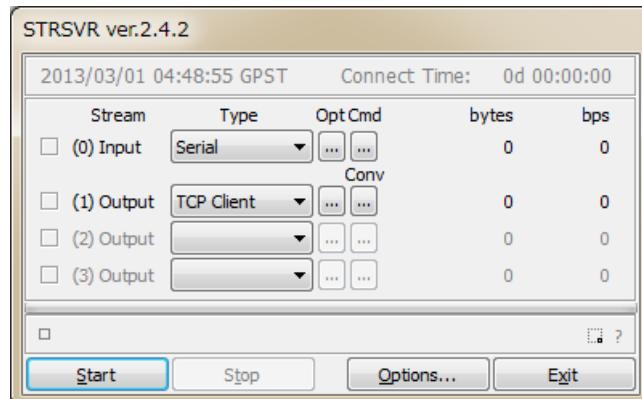


Figure 3.3-1 Main Window of STRSVR

- (2) To configure the input stream, select the stream type with pull down menu at "(0) Input". Selectable stream types are Serial, TCP Client, TCP Server, NTRIP Client, or File. The stream options or the startup/shutdown command can be set as well as Input Streams for RTKNAVI.
- (3) To configure the output streams, select the stream type with pull down menu at (1) Output, (2) Output or (3) Output. The setting for the output streams are Serial, TCP Client, TCP Server, NTRIP Server, NTRIP Caster, or File
- (4) In version 2.4.2, the stream format conversion function is added. To use the function, push Conv

button right of the pull down menu of the output stream. You can see "Conversion Options" dialog.

To enable the stream format conversion function, check the upper left checkbox in the dialog and select input and output format by the pull down menus. Current version supports the following input and output formats.

(a) Input: RTCM3, RTCM2, NovAtel OEM6, NovAtel OEM3, u-blox, Superstar II, Hemisphere, SkyTraq, GW10, Javad, NVS BINR and BINEX

(b) Output: RTCM 3 (RTCM 2 is not supported yet)

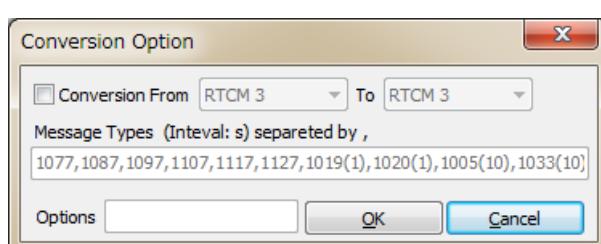


Figure 3.3-2 Conversion Option Dialog of STRSVR

Output messages shall be specified in the Message Types field in the dialog as the form:

**nnnn(ss), nnnn(ss), nnnn(ss), ...**

Specify the message types as the fields **nnnn** and the message intervals as the fields **(ss)** in seconds. The message interval can be omitted. In this case, the message interval is determined by the input message interval. The following table shows all of the supported output RTCM messages. For antenna info messages, the fields are given by the "Options" dialog. Message input options can also be specified in the "Options" field. Refer D.5 for the receiver dependent options for details.

#### Supported output RTCM 3 message type

TYPE	GPS	GLONASS	Galileo	QZSS	BeiDou	SBAS
OBS C-L1 : 1001	1009	-	-	-	-	-
F-L1 : 1002	1010	-	-	-	-	-
C-L12 : 1003	1011	-	-	-	-	-
F-L12 : 1004	1012	-	-	-	-	-
NAV : 1019	1020	1045*	1044*	-	-	-
	-	1046*	-	-	-	-
MSM 1 : 1071	1081	1091	1111*	1121*	1101*	
2 : 1072	1082	1092	1112*	1122*	1102*	
3 : 1073	1083	1093	1113*	1123*	1103*	
4 : 1074	1084	1094	1114*	1124*	1104*	
5 : 1075	1085	1095	1115*	1125*	1105*	
6 : 1076	1086	1096	1116*	1126*	1106*	
7 : 1077	1087	1097	1117*	1127*	1107*	
ANT INFO : 1005 1006 1007 1008 1033						

\* draft version of RTCM 3 messages

- (5) Push **Start** button in the main window. The communication status is shown in the message area lower center of the main window. Status indicators left side of the main window also shows the communication status. The indicator colors means: **Orange**: waiting connection, **Dark-Green**: connected, **Light-Green**: data active, **Red**: error. Total data amount (bytes) and data rate (bps) of the input and output streams are also shown in right side. To stop the communication, push **Stop** button.
- (6) By pushing **Options...** button, you can set the communication options with the "Options" dialog. To send NMEA GPGGA message to the server connected the input stream, check "NMEA Request Cycle" and set the request cycle (ms) and latitude/longitude in the messages. To connect an external NTRIP caster from the inside of the firewall via a HTTP-proxy server, you can input the address and the port number as the form **<address>:<port>** in the "HTTP/NTRIP Proxy" field. The other fields in the dialog are for antenna and station information messages to be generated in case of using the format conversion feature and sending antenna information messages.

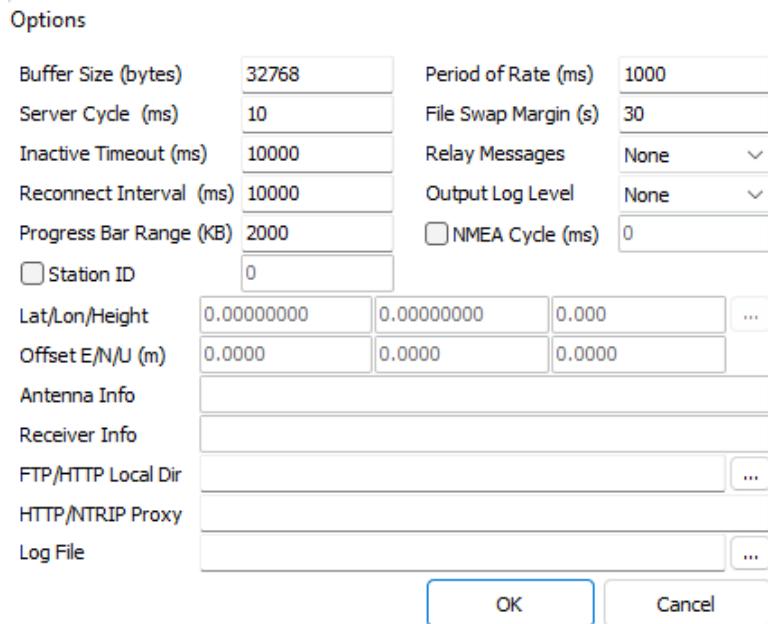


Figure 3.3-3 Options Dialog of STRSVR

### 3.4 Post-Processing Analysis with RTKPOST

RTKLIB contains a post processing analysis AP RTKPOST. RTKPOST inputs the standard RINEX 2.10, 2.11, 2.12, 3.00, 3.01, 3.02 (draft) observation data and navigation message files (GPS, GLONASS, Galileo, QZSS, BeiDou and SBAS) and can computes the positioning solutions by various positioning modes including Single-point, DGPS/DGNSS, Kinematic, Static, PPP-Kinematic and PPP-Static.

- (1) Execute the binary AP file `<install dir>\rtklib_<ver>\bin\rtkpost.exe`. You can see the main window of RTKPOST. You can execute the binary AP file:

`<install dir>\rtklib_<ver>\bin\rtkpost_mkl.exe` instead, which is a version of RTKPOST linking the Intel MKL library for fast matrix computation.

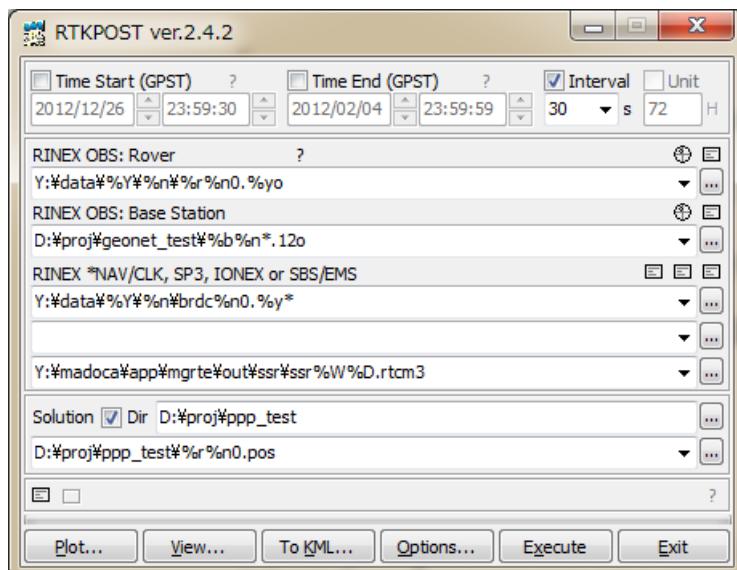


Figure 3.4-1 Main Window of RTKPOST

- (2) Input the RINEX observation data file path of the rover receiver in the text field "RINEX OBS(: Rover)". Fill in the file path or select a file using the file selection dialog shown by pushing  button. You can use the compressed file by GZIP [57] (`.gz`), COMPRESS (`.z`) or Hatanaka-Compression [58] (`.yyd`) for the RINEX observation data. If the compression file or not is recognized by the file extension. If a wild-card (\*) is included in the file path, the wild-card is expanded and the multiple files are read.
- (3) If you process RINEX data in the relative positioning modes as: DGPS/DGNSS, Kinematic, Static, Moving-Base or Fixed, you have to input the second file path of the base-station receiver in the "RINEX OBS: Base Station" field in addition to the rover observation data file.

(4) You also have to input the path of RINEX navigation message files of GPS, GLONASS, Galileo, QZSS and SBAS in the "RINEX \*NAV/CLK, SP3, IONEX or SBS/ EMS" field. If you leave first and second field blank, the observation data file path with the extension replaced by **.\*nav** (**.obs**) or **.yyN**, **.yyG**, **.yyH**, **.yyQ** and **.yyP** (**.yyO**) is used for the navigation message files of GPS, GLONASS, Galileo, QZSS, BeiDou and SBAS. If a wild-card (\*) is included in the file path, the wild-card is expanded and the multiple files are used like observation data files. To use precise ephemeris and clock for PPP-Kinematic, PPP-Static or PPP-Fixed mode, you can input a SP3-c (for precise satellite ephemeris and clock) or RINEX CLK (for precise satellite clock) file path in the field. You can input an IONEX 1.0 file path for ionospheric VTEC grid corrections. For SBAS corrections, you can input a SBAS message log file path as RTKLIB format or EMS (EGNOS message server) 2.0 format file. You can also include wild-cards (\*) in these file paths. The wild-cards are expanded and multiple files are used. You can input SSR (state space representation) corrections as RTCM 3 messages in a input file field. The formats of these input files are recognized by their extensions as follows:

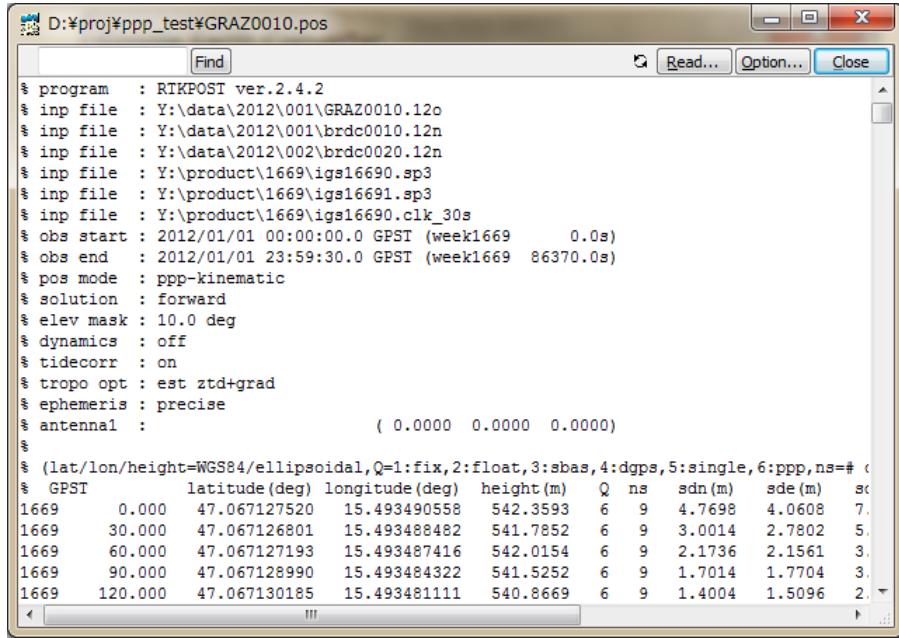
- (a) **.sp3**, **.SP3**, **.eph**, **.EPH** : SP3-c precise ephemeris file [22]
- (b) **.sbs**, **.SBS**, **.ems**, **.EMS** : SBAS message log file (Appendix B.2 and [26])
- (c) **.rtcm3**, **.RTCM3** : RTCM 3 SSR correction message file [18]
- (d) **\*i**, **\*I** : IONEX VTEC grid data file [24]
- (e) **others** : RINEX OBS, NAV or CLK (automatically recognized) [9]-[15]

- (5) Input the output file path in the text field "Solution". The field is automatically set as the first input file path with the extension replaced by **.pos** or **.nmea**. If you check "Dir" and fill in the field, the output directory is set to the specified directory. You can modify the output file path manually by editing the field content.
- (6) Push **Options...** button to set the processing options. For the detailed options for RTKPOST, refer 3.5 Configure Positioning Options for RTKNAVI and RTKPOST. You can set the start time or end time by checking and setting Time Start (GPST) or Time End (GPST) field in the main window. You also set the time interval by checking and setting the "Interval" field. With the **[?]** button, the input time in GPS time can be converted to UTC, GPS Week/TOW, Day of Year, Day of Week, Time of Day and Leap Seconds.



Figure 3.4-2 Time Dialog of RTKPOST

- (7) If you check both of the "Time Start" and "Time End" fields, you can check "Unit" for multiple session analysis. If the "Unit" field checked and set the "Unit" time in hours, the analysis session is separated to multiple sessions for the unit time. To avoid overwriting the previous output file, the output file path has to contain the keyword replaced according to the session time. For the details of the keyword replacement in the input or output file paths, refer 3.5 Configure Positioning Options for RTKNAVI and RTKPOST.
- (8) Push **Execute** button to start the analysis. The processing status is shown in the status message field lower center in the main window. When you see "**done**" message here, the analysis is completed. If you want to stop the processing on the way, push **Abort** button.
- (9) After completing the analysis, by pushing **View...** button, you can display the content of the output file by "Text Viewer". You can reload the output file by pushing **Reload** button in the "Text Viewer" window. To close the window, push **Close** button. You can configure "Text Viewer" options by pushing **Options...** button. You can also search strings in the text by using **Find** button.



```

D:\proj\ppp_test\GRAZ0010.pos

$ program : RTKPOST ver.2.4.2
$ inp file : Y:\data\2012\001\GRAZ0010.12o
$ inp file : Y:\data\2012\001\brdc0010.12n
$ inp file : Y:\data\2012\002\brdc0020.12n
$ inp file : Y:\product\1669\igs16690.sp3
$ inp file : Y:\product\1669\igs16691.sp3
$ inp file : Y:\product\1669\igs16690.clk_30s
$ obs start : 2012/01/01 00:00:00.0 GPST (week1669 0.0s)
$ obs end : 2012/01/01 23:59:30.0 GPST (week1669 86370.0s)
$ pos mode : ppp-kinematic
$ solution : forward
$ elev mask : 10.0 deg
$ dynamics : off
$ tidecorr : on
$ tropo opt : est ztd+grad
$ ephemeris : precise
$ antennal : ( 0.0000 0.0000 0.0000)
$

# (lat/lon/height=WGS84/ellipsoidal,Q=1:fix,2:float,3:sbas,4:dgps,5:single,6:ppp,ns=#)
# GPST      latitude(deg) longitude(deg) height(m) Q ns sdn(m) sde(m) sc
1669     0.000    47.067127520 15.493490558 542.3593   6   9   4.7698   4.0608   7,
1669    30.000    47.067126801 15.493488482 541.7852   6   9   3.0014   2.7802   5,
1669    60.000    47.067127193 15.493487416 542.0154   6   9   2.1736   2.1561   3,
1669   90.000    47.067128990 15.493484322 541.5252   6   9   1.7014   1.7704   3,
1669  120.000    47.067130185 15.493481111 540.8669   6   9   1.4004   1.5096   2,

```

Figure 3.4-3 Text Viewer showing Solutions by RTKPOST

- (10) By pushing **Plot...** button, you can also plot the result with RTKPLOT. Refer 3.7 View and Plot Solutions and Observation Data with RTKPLOT for details.

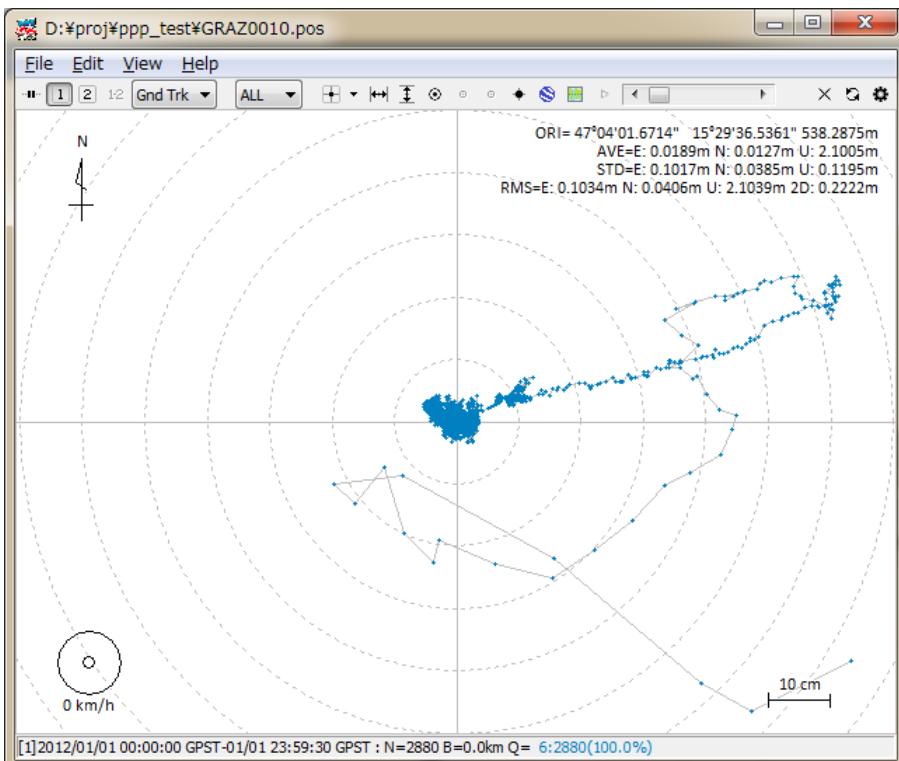


Figure 3.4-4 RTKPLOT Window executed by RTKPOST

- (11) By pushing **KML/GPX...** button, the output file can be converted to Google Earth KML file with the "Google Earth Converter" dialog. Set or select the options and push **Convert** button in the dialog. You can launch Google Earth with the generated KML/KMZ file by pushing **Google Earth** button. To specify the Google Earth execution file, configure "Options" - "Files" - "Google Earth Exe File".

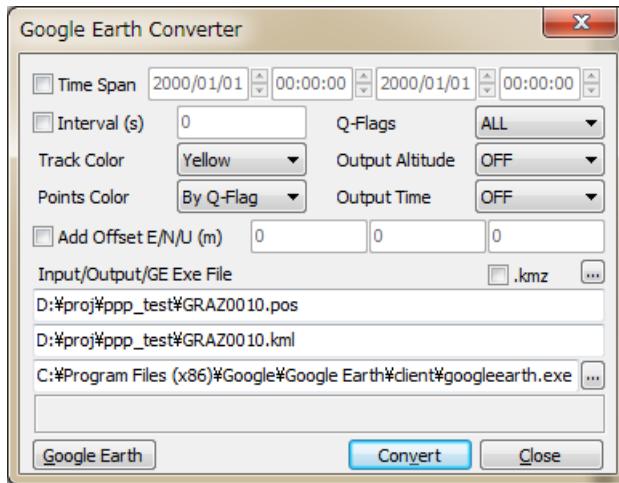


Figure 3.4-5 Google Earth Converter Dialog of RTKPOST

- (12) With **⊕** button in the main window, you can view and plot the input observation data RTKPLOT. You can also display the contents of the input files with Text Viewer by pushing **...** button.
- (13) In case of output solution statistics or debug trace as the processing options, push **...** buttons lower left of the window to view the solution statistics file or the debug trace file. To check processing error or warning in case of improper results, set Debug Trace in "Options" - "Output" dialog to "Level 2" (trace ERROR and WARNING) and see the output debug trace file.

### 3.5 Configure Positioning Options for RTKNAVI and RTKPOST

By pushing **Options...** button in the main windows of RTKNAVI or RTKPOST, you can set the positioning options. Selectable or changeable positioning options are as follows. These options can be saved to the configuration file by pushing **Save** button on the dialog and select the file path. The options can be loaded from a configuration file by pushing **Load** button and selecting a configuration file. For the configuration file, refer B.4. The keywords which can be included the configuration file are also shown in the following tables. The models specified in these options are also explained in Appendix E for details.

#### (1) Setting 1

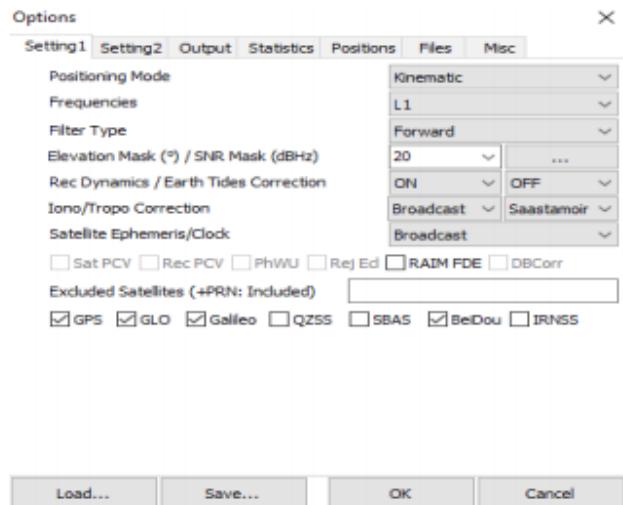


Figure 3.5-1 Options Dialog (Setting 1) of RTKNAVI and RTKPOST

Item	Descriptions	Configuration File	Notes
Positioning Mode	<p>Set positioning mode</p> <ul style="list-style-type: none"> <li>- Single : Single point positioning or SBAS DGPS</li> <li>- DGPS/DGNSS : Code-based differential GPS</li> <li>- Static : Carrier-based Static positioning</li> <li>- Static-start: Static till first fix, then Kinematic</li> <li>- Kinematic: Carrier-based Kinematic positioning</li> <li>- Moving-Base: Moving baseline</li> <li>- Fixed: Rover receiver position is fixed *</li> <li>- PPP Kinematic: Precise Point Positioning with kinematic mode</li> <li>- PPP Static: Precise Point Positioning with static mode</li> <li>- PPP Fixed: Rover receiver position is fixed with PPP mode *</li> </ul>	<b>pos1-posmode</b>	*
Frequencies	<p>Set used carrier frequencies</p> <ul style="list-style-type: none"> <li>- L1: L1 Single frequency: L1,R1,E1,B1</li> <li>- L1+L2: L1 and L2 Dual-frequency: L1/L2, R1/R2, E1/E5b, B1/B2</li> <li>- L1+L2+L5: L1, L2 and L5 Triple-frequency L1/L2/L5, R1/R2, E1/E5b/E5a, B1/B2</li> <li>- L1+L2+L5+L6: Experimental, not fully supported</li> </ul>	<b>pos1-frequency</b>	N/A to Single, PPP-* modes
Filter Type	<p>Set filter type</p> <ul style="list-style-type: none"> <li>- Forward: Forward filter solution</li> <li>- Backward: Backward filter solution *</li> <li>- Combined: Smoother combined solution with forward and backward filter solutions *, phase bias states reset between forward and backward solutions.</li> <li>- Combined -no phase reset: Smoother combined solution with forward and backward filter solutions *, phase bias states not reset between forward and backward solutions.</li> </ul>	<b>pos1-solttype</b>	*
Elevation Mask	Set elevation mask angle in degree.	<b>pos1-elmask</b>	
SNR Mask	Set SNR mask. Push ... button to show the "SNR Mask" dialog. Set SNR thresholds to reject satellite signals for each 5 deg elevation bins in the dialog. If both of "Rover" and "Base Station" are unchecked, these SNR masks are not applied.	<b>pos1-snrmask_r, snrmask_b, snrmask_L1, snrmask_L2, snrmask_L5</b>	
Rec Dynamics	<p>Set the dynamics model of the rover receiver.</p> <ul style="list-style-type: none"> <li>- OFF: Dynamics is not used</li> <li>- ON: Receiver velocity and acceleration are estimated.</li> </ul> <p>The receiver position is predicted with the estimated velocity and acceleration.</p>	<b>pos1-dynamics</b>	Only applicable to DGPS/DGNSS or Kinematic modes

Item	Descriptions	Configuration File	Notes
Earth Tides Correction	<p>Set whether earth tides correction is applied or not</p> <ul style="list-style-type: none"> <li>- OFF: Not apply earth tides correction</li> <li>- Solid: Apply solid earth tides correction</li> <li>- Solid/OTL: Apply solid earth tides, OTL (ocean tide loading) and pole tide corrections. *</li> </ul> <p>To apply OTL correction, set the OTL coefficients file path in "Ocean Loading BLQ Format" in the "Files" tab and the marker name have to be include in the input RINEX file to select the station in BLQ file.</p> <p>To apply pole tide, set ERP (earth rotation parameter) file path in "EOP Data File" in the "Files" tab.</p>	<b>pos1-tidecorr</b>	N/A to Single mode *
Ionosphere Correction	<p>Set ionospheric correction options. If you set the parameter Estimated. Vertical ionospheric delay for each satellite) are estimated. For long baseline analysis, ionosphere estimation is effective to suppress ionosphere delay effects.</p> <ul style="list-style-type: none"> <li>- OFF : Not apply ionospheric correction</li> <li>- Broadcast: Apply broadcast ionospheric model</li> <li>- SBAS: Apply SBAS ionospheric model</li> <li>- Iono-Free LC: Ionosphere-free linear combination with dual frequency (L1-L2 for GPS/ GLONASS/ QZSS or L1-L5 for Galileo) measurements is used for ionospheric correction</li> <li>- Estimate STEC : Estimate ionospheric parameter STEC (slant total electron content) *</li> <li>- IONEX TEC: Use IONEX TEC grid data</li> <li>- QZSS Broadcast: Apply broadcast ionosphere model provided by QZSS</li> <li>- QZSS LEX: (reserved)</li> </ul>	<b>pos1-ionoopt</b>	* N/A to Single, PPP-* modes
Troposphere Correction	<p>Set whether tropospheric parameters (zenith total delay at rover and base-station positions) are estimated or not.</p> <ul style="list-style-type: none"> <li>- OFF : Not apply troposphere correction</li> <li>- Saastamoinen: Apply Saastamoinen model</li> <li>- SBAS: Apply SBAS tropospheric model (MOPS)</li> <li>- Estimate ZTD: Estimate ZTD (zenith total delay) parameters as EKF states *</li> <li>- Estimate ZTD+Grad: Estimate ZTD and horizontal gradient parameters as EKF states *</li> </ul>	<b>pos1-tropopt</b>	* N/A to Single mode.

Item	Descriptions	Configuration File	Notes
Satellite Ephemeris/ Clock	Set the type of satellite ephemeris. - Broadcast : Use broadcast ephemeris - Precise : Use precise ephemeris * - Broadcast+SBAS: Broadcast ephemeris with SBAS long-term and fast correction - Broadcast+SSR APC: Broadcast ephemeris with RTCM SSR correction (antenna phase center value) - Broadcast+SSR CoM: Broadcast ephemeris with RTCM SSR correction (satellite center of mass value) - QZSS LEX: (reserved)	<b>pos1- sateph</b>	
Sat PCV	Set whether the satellite antenna PCV (phase center variation) model is used or not. To use the feature, set "Satellite Antenna PCV File" in "Files" tab.	<b>pos1- posopt1</b>	N/A to Single mode
Rec PCV	Set whether the receiver antenna PCV model is used or not. To use the feature, set "Receiver Antenna PCV File" in "Files" tab.	<b>pos1- posopt2</b>	N/A to Single mode
PhWindup	Set whether the phase windup correction for PPP modes is applied or not.	<b>pos1- posopt3</b>	Only applicable to PPP-* modes.
Reject Ecl	Set whether the GPS Block IIA satellites in eclipse are excluded or not. The eclipsing Block IIA satellites often degrade the PPP solutions due to unpredicted behavior of yaw-attitude.	<b>pos1- posopt4</b>	Only applicable to PPP-* modes.
RAIM FDE	Set whether RAIM (receiver autonomous integrity monitoring) FDE (fault detection and exclusion) feature is enabled or not. In case of RAIM FDE enabled, a satellite is excluded if SSE (sum of squared errors) of residuals is over a threshold. The excluded satellite is selected to indicate the minimum SSE.	<b>pos1- posopt5</b>	
DBCorr	Day boundary clock jump correction	<b>pos1- posopt6</b>	Only applicable to PPP-* modes.
Excluded Satellites (+PRN: Included)	Set the excluded satellites for positioning. Fill in the PRN numbers of the satellites separated by spaces. For GLONASS, Galileo, QZSS, BeiDou and SBAS, use Rnn, Enn, Jnn, Cnn and Snn, respectively (nn: satellite PRN or slot number). If "+" is added to the head of the satellite ID, the satellite is included for positioning even if the satellite is unhealthy.	<b>pos1- exclsats</b>	

Item	Descriptions	Configuration File	Notes
Navigation System	Check used navigation satellite systems. If unchecked, satellites of the system are not used for positioning. - GPS - GLONASS - Galileo - QZSS - SBAS - BeiDou - IRNSS	<b>pos1-navsys</b>	

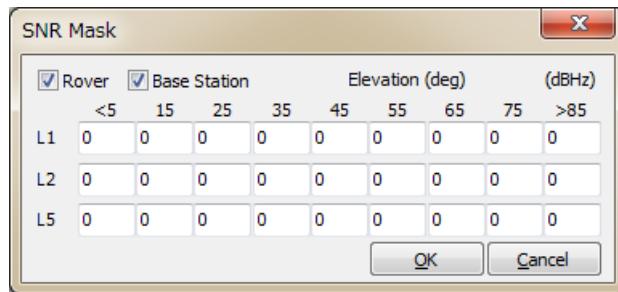


Figure 3.5-2 SNR Mask Dialog of RTKNAVI and RTKPOST Options

## (2) Setting 2

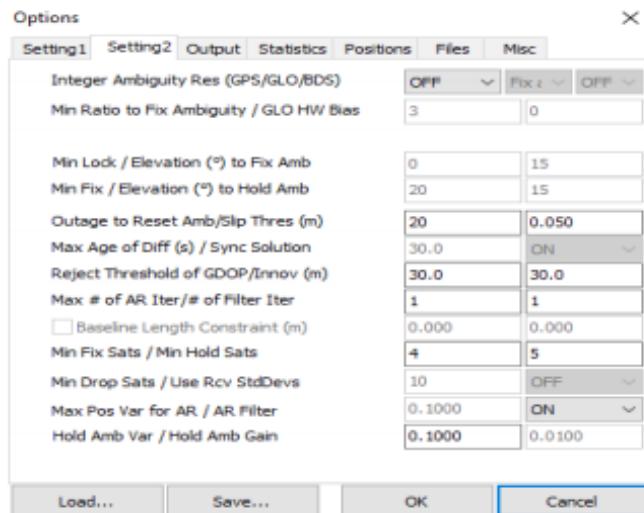


Figure 3.5-2 Options Dialog (Setting 2) of RTKNAVI and RTKPOST

Item	Descriptions	Configuration File	Notes
Integer Ambiguity Resolution (GPS)	<p>Set the strategy of integer ambiguity resolution for GPS and Galileo</p> <ul style="list-style-type: none"> <li>- OFF : No ambiguity resolution</li> <li>- Continuous : Continuously static integer ambiguities are estimated and resolved *</li> <li>- Instantaneous : Integer ambiguity is estimated and resolved by epoch-by-epoch basis *</li> <li>- Fix and Hold : Continuously static integer ambiguities are estimated and resolved. If the validation OK, the ambiguities are constrained to the resolved values. *</li> <li>- PPP-AR : Not supported</li> </ul>	<b>pos2-armode</b>	Default: Continuous Not applicable to Single mode. * Only applicable to Kinematic, Static, Moving-baseline and Fixed modes.
Integer Ambiguity Resolution (GLO)	<p>Set the strategy of GLONASS integer ambiguity resolution</p> <ul style="list-style-type: none"> <li>- OFF: Ambiguities are not fixed.</li> <li>- ON: Ambiguities are fixed. Usually the ambiguity of only the same types receiver pair for the rover and the base station can be fixed. If he different receiver types have IFB (inter-frequency bias), they cannot be canceled by DD.</li> <li>- Fix and Hold: Nulls out IF biases after first fix-and-hold of GPS satellites</li> <li>- Autocal: Receiver IF biases auto-calibrate but require reasonably accurate initial estimates (see GLO HW Bias)</li> </ul>	<b>pos2-gloarmode</b>	Default: ON Only applicable to Kinematic, Static, Moving-baseline and Fixed modes.
Integer Ambiguity Resolution (BDS)	Set the strategy of integer ambiguity resolution for BeiDou (ON/OFF)	<b>pos2-bdsarmode</b>	Default: OFF
Ratio to Fix Ambiguity (Min/Nom/M ax)	<ul style="list-style-type: none"> <li>- Set the integer ambiguity validation threshold for "ratio-test", which uses the ratio of squared residuals of the best integer vector to the second-best vector.</li> <li>- If min and max values are set equal to the nominal value then the ratio is fixed, else the ratio will be adjusted by the number of satellites. The nominal value is used for 8 satellite pairs and is adjusted lower as the number of satellites increases and higher as it decreases .</li> </ul>	<b>pos2-arthresmin</b> <b>pos2-arthres</b> <b>pos2-arthresmax</b>	Default value: 3.0
GLO HW Bias	Set relative GLONASS hardware bias in meters per frequency slot. Used only when pos2-gloarmode is set to "autocal" and is used to specify the interchannel bias between two different receiver manufacturers	<b>pos2-arthres2</b>	2 Default value: 0.0
Min Lock / Elevation to Fix Ambiguity	Set the minimum lock count and the minimum elevation angle (deg) to fix integer ambiguity. If the lock count or the elevation angle is less than the value, the ambiguity is excluded from the fixed integer vector.	<b>pos2-arlockcnt</b> , <b>arelmask</b>	Default value: 0, 0

Item	Descriptions	Configuration File	Notes
Min Fix / Elevation to Hold Ambiguity	If you select "Fix and Hold" mode for Integer Ambiguity Resolution, set the minimum fix count and the minimum elevation angle (deg) to hold ambiguity.	<b>pos2-arminfix, elmaskhold</b>	Default value: 10, 0
Slip Threshs: Doppler (Hz) / Geom-Free (m)	Set the cycle-slip threshold (Hz) of difference between doppler and carrier phase difference between epochs (0 = disabled)/ Set the cycle-slip threshold (m) of geometry-free LC carrier-phase difference between epochs.	<b>pos2-dopthres, pos2-sliphres</b>	Default value: 0, 0.05
Max Age of Diff / Outs to Reset Amb	Set the maximum value of age of differential (s) between the rover and the base station. / Set the outage count to reset ambiguity. If the data outage count is over the value, the estimated ambiguity is reset to the initial value.	<b>pos2-maxage, pos2-aroutcnt</b>	Default value: 30, 5
Reject Threshold of GDOP/ Innov.	Set the reject threshold of GDOP and kalman filter innovations (m). If the GDOP or the carrier phase innovation is over the value, the observable is excluded for the estimation process as an outlier. The innovation threshold is multiplied by the carrier phase/pseudorange error ratio to generate the threshold for pseudorange innovations.	<b>pos2-rejgdop, rejionno</b>	Default value: 30, 1.0
# of Filter Iter	Set the number of iteration in the measurement update of the estimation filter. If the baseline length is very short like 1 m, the iteration may be effective to handle the nonlinearity of measurement equation.	<b>pos2-niter</b>	Default value: 1
Sync Solution	Set time synchronization mode of solutions: - OFF: Minimum latency mode. The solution is output soon after rover data input. The delayed base station or correction data are extrapolated to the rover time.- ON: Matched solution mode. The solution is output after both rover data and base station or correction data prepared. The solution time may be behind the rover time with a certain delay.	<b>pos2-syncsol</b>	N/A to RTKPOST and Single mode Default value: OFF
Baseline Length Constraint	If Moving-Base mode, check and set the constraint of the baseline length. Fill in the length in m and the standard deviation (m) of the constraint.	<b>pos2-baselens, basesig</b>	Default value: 0.0
Min Fix Sats	Minimum ambiguities required for fix /	<b>pos2-minfixsat,</b>	Default value:
Min Hold Sats	Minimum ambiguities required for hold	<b>pos2-minholdsats</b>	4, 5
Min Drop Sats	Minimum ambiguities required to enable exclusion of a single satellite from ambiguity resolution each epoch	<b>Pos2-mindropsats</b>	Default value: 10
Max Pos Var for AR	Ambiguity resolution delayed until variance of the position states reaches this threshold to minimize false fixes	<b>pos2-arthres1</b>	Default value: 0.25
AR Filter	Require AR qualification of new or returning satellites after cycle slip (ON/OFF)	<b>pos2-arfilter</b>	Default value: ON
Hold Amb Var	Set variance for ambiguity hold feedback (acts as inverse ambiguity tracking gain)	<b>pos2-varholdamb</b>	Default value: 0.1

Item	Descriptions	Configuration File	Notes
Hold Amb Gain	Gain for GLONASS IF bias nulls when using GLONASS "Fix and Hold" AR mode	<b>pos2-gainholdamb</b>	Default value: 0.01

### (3) Output

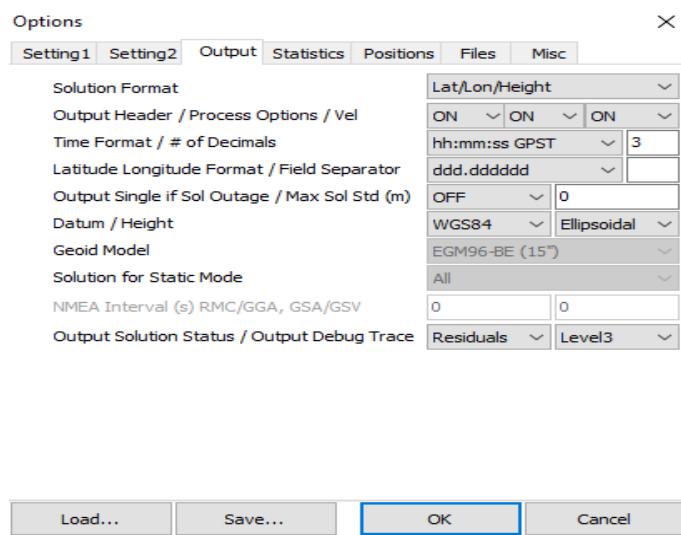


Figure 3.5-3 Options Dialog (Output) of RTKNAVI and RTKPOST

Item	Descriptions	Configuration File	Notes
Solution Format	Set the output solution format. - Lat/Lon/Height : Latitude, longitude and height - X/Y/Z-ECEF : X/Y/Z components of ECEF coordinates - E/N/U-Baseline: E/N/U components of baseline vector - NMEA0183 : NMEA GPRMC, GPGGA, GPGSA, GLGSA, GAGSA, GPGSV, GLGSV and GAGSV	<b>out-solformat</b>	For RTKNAVI, specify options as Output Streams setting.
Output Header	Set whether the header is output or not.	<b>out-outhead</b>	N/A to NMEA
Process Options	Set whether the processing options are output or not.	<b>out-outopt</b>	RTKPOST only N/A to NMEA
Vel	Set whether the velocity outputs are enabled or not	<b>out-outvel</b>	
Time Format	Set the format of time - ssssssss.sss GPST : GPS week and time of week - hh:mm:ss GPST : yyyy/mm/dd hh:mm:ss GPST - hh:mm:ss UTC : yyyy/mm/dd hh:mm:ss UTC - hh:mm:ss JST : yyyy/mm/dd hh:mm:ss JST	<b>out-timesys, timeform</b>	N/A to NMEA
# of Decimals	Set number of decimals in the time format	<b>out-timendec</b>	N/A to NMEA

Item	Descriptions	Configuration File	Notes
Latitude/ Longitude Format	Set the formats of latitude and longitude if the solution format is set to Lat/Lon/Height. - ddd.dddddddd : Degree - ddd mm ss.sss : Degree minute second	<b>out-degform</b>	N/A to NMEA
Field Separator	Set the separator for fields.	<b>out-fieldsep</b>	N/A to NMEA
Datum	Set the datum if the solution format option is set to Lat/Lon/Height. - WGS84 : WGS84 datum - Tokyo : Tokyo datum (current version supports only WGS84)	-	
Height	Set the type of height. - Ellipsoidal : Ellipsoidal height - Geodetic : Geodetic height	<b>out-height</b>	
Geoid Model	Set the geoid model if the Height option is set to Geodetic. - Internal: Internal geoid model - EGM96-BE (15") : EGM96 (15" x 15" grid) *1 - EGM2008-SE (2.5") : EGM2008 (2.5 x 2.5" grid) *2 - EGM2008-SE (1") : EGM2008 (1 x 1" grid) *2 - GSI2000 (1x1.5") : GSI2000 (1x1.5" grid) *3 If using external geoid model, specify the geoid file path in "Files" tab.	<b>out-geoid</b>	
Solution for Static Mode	Set the solution type for Static or PPP-Static mode - All: all solutions for the processing period are outputted - Single: Only one solution for the processing period is output. The time of solution is first epoch in the processing period.	<b>out-solstatic</b>	RTKPOST only
NMEA Interval (s) RMC/GGA	Set the output interval of NMEA GPRMC, GPGGA messages	<b>out-nmeaintv1</b>	RTKNAVI only
NMEA Interval (s) GSA/GSV	Set the output interval of NMEA GPGSA, GLGSA, GAGSA, GPGSV, GLGSV, GAGSV messages	<b>out-nmeaintv2</b>	RTKNAVI only
Output Solution Status	Set the output level of the solution status file. The solution status file contains estimated states and residuals. The solution status file is created in the current directory (RTKNAVI) or in the output file directory (RTKPOST).	<b>out-outstat</b>	
Output Debug Trace	Set the output level of debug trace file. If setting OFF, any debug trace file is not output. The debug trace file is created in the current directory (RTKNAVI) or in the output file directory (RTKPOST).	-	

\*1 WW15MGH.DAC (<http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm96/binary/binarygeoid.html>)

\*2 Und\_min1x1\_egm2008\_isw=82\_WGS84\_TideFree\_SE,

Und\_min2.5x2.5\_egm2008\_isw=82\_WGS84\_TideFree\_SE

([http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/egm08\\_wgs84.html](http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm2008/egm08_wgs84.html))\*3 gsigeome.ver4 (<http://vldb.gsi.go.jp/sokuchi/geoid/download/down.html>)

## (4) Statistics

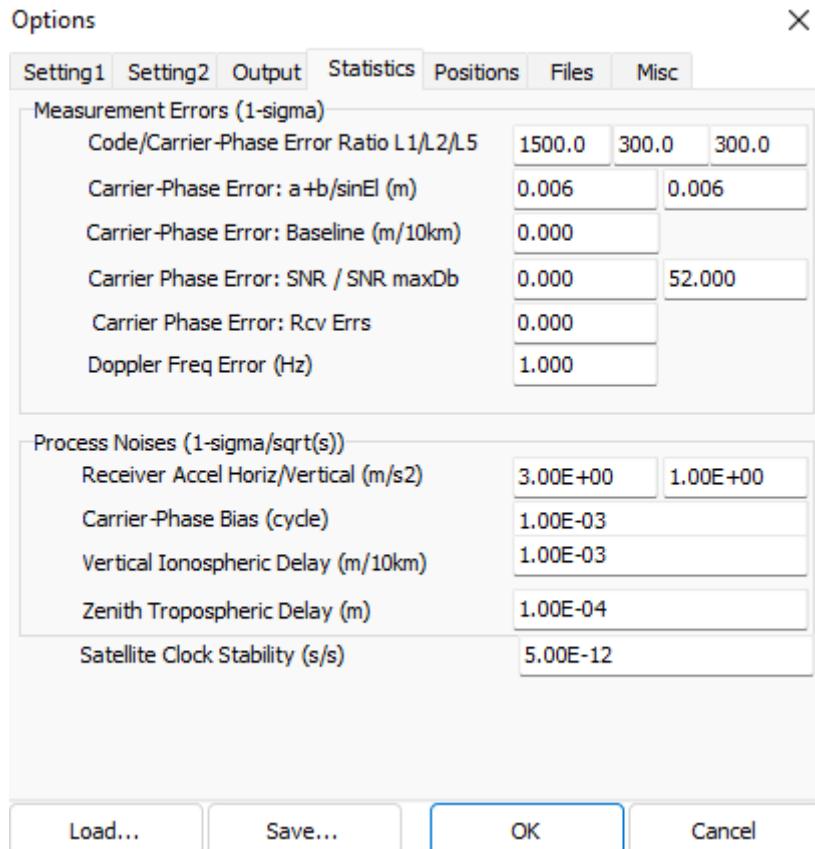


Figure 3.5-4 Options Dialog (Statistics) of RTKNAVI and RTKPOST

Item	Descriptions	Configuration File	Notes
<b>Measurement Errors</b>			
Code/Carrier-Phase Error Rate L1/L2/L5	Set the ratio of standard deviations of pseudorange errors to carrier-phase errors for L1, L2, and L5	<b>stats-eratio1</b> , <b>eratio2</b> , <b>eratio5</b>	Default value: 300, 300,300
Carrier-Phase Error: a	Set the base term of carrier-phase error standard deviation (m).	<b>stats-errphase</b>	Default value: 0.003
Carrier-Phase Error: b/sinEl	Set the elevation dependent term of carrier-phase error standard deviation (m/sin(el)).	<b>stats-errphaseel</b>	Default value: 0.003
Carrier-Phase Error/Baseline	Set the baseline-length dependent term of carrier-phase error standard deviation (m/10km).	<b>stats-errphasebl</b>	Default value: 0
Carrier-Phase Error: SNR: e / SNR Max	Set the SNR dependent terms of carrier-phase error standard deviation. $e * 10^{(0.1*(SNR\_MAX - snr))}$	<b>stats-errsnr</b> <b>snrmax</b>	Default value: 0, 0
Carrier-Phase Error: RCV Errs	Set the receiver uncertainty dependent term of carrier-phase error standard deviation. (u-blox only)	<b>Stats-errrcv</b>	Default value: 0
Doppler Frequency	Set the standard deviation of Doppler errors (Hz)	<b>stats-errdoppler</b>	Default value: 1

Item	Descriptions	Configuration File	Notes
<b>Process Noises</b>			
Receiver Accel	Set the process noise standard deviation of the receiver acceleration as the horizontal or vertical component. (m/s <sup>2</sup> /sqrt(s)). If Receiver Dynamics is set to OFF, they are not used.	<b>stats-prnaccelh,</b> <b>stats-prnaccelv</b>	Default value: 1 and 0.1
Carrier-Phase Bias	Set the process noise standard deviation of carrier-phase bias (ambiguity) (cycle/sqrt(s)).	<b>stats-prnbias</b>	Default value: 1E-4
Vertical Ionospheric Delay	Set the process noise standard deviation of vertical ionospheric delay per 10 km baseline (m/sqrt(s)).	<b>stats-prniono</b>	Default value: 1E-3
Zenith Tropospheric Delay	Set the process noise standard deviation of zenith tropospheric delay (m/sqrt(s)).	<b>stats-prntrop</b>	Default value: 1E-4
Satellite Clock Stability	Set the satellite clock stability (s/s). The value is used for interpolation of base-station observables.	<b>stats-clkstab</b>	Default value: 5E-12

## (5) Positions

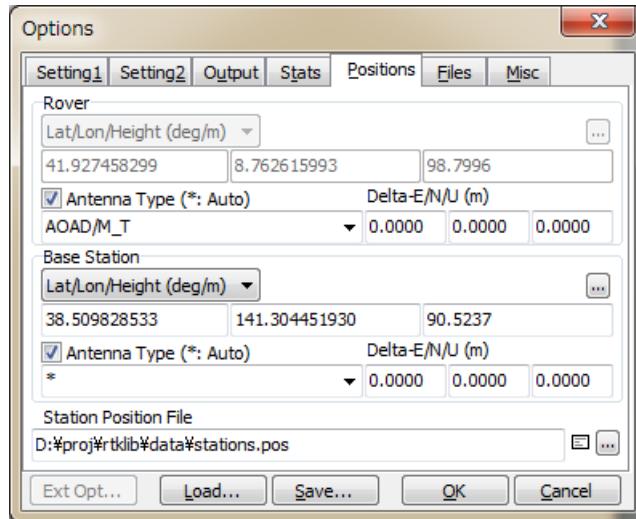


Figure 3.5-5 Options Dialog (Positions) of RTKNAVI and RTKPOST

Item	Descriptions	Configuration File	Notes
Rover			
Lat/Lon/Height (deg/m)	Set the position of the rover antenna if the rover antenna is fixed. See the same field for Base Station.	<b>ant1-postype,</b> <b>pos1,</b> <b>pos2, pos3</b>	

Item	Descriptions	Configuration File	Notes
Antenna Type	Select the type of the rover antenna. To select the antenna type, set the Receiver Antenna PCV File path in Files. If "*" is used, antenna type and antenna delta are recognized by the antenna information of RINEX OBS header (RTKPOST) or RTCM antenna information (RTKNAVI).	<b>ant1-</b> <b>anttype</b>	
Delta-E/N/U	Set the delta position of the rover antenna as the E/N/U offsets of ARP (antenna reference point) position with refer to the marker (m).	<b>ant1-</b> <b>antdel1e,</b> <b>antdel1n,</b> <b>antdel1u</b>	
<b>Base Station</b>			
Lat/Lon/Height (deg/m)	Set the position of the base-station antenna. - Lat/Lon/Height (deg/m): Latitude/longitude/height in degree and m - Lat/Lon/Height (dms/m): Latitude/longitude/height in degree/minute/second and m - X/Y/Z-ECEF (m): X/Y/Z components in ECEF frame. - RTCM Station Position: Use the antenna position included in RTCM messages * - Average of Single-Pos: Use the average of single point solutions ** - Get from Position File: Use the position in the position file. The station is searched by using the head 4-character ID of the rover observation data file path. ** - RINEX Header Position: Use the approximate position in RINEX OBS header. **	<b>ant2-</b> <b>postype,</b> <b>pos1,</b> <b>pos2, pos3</b>	Height is specified as ellipsoidal height * RTKNAVI only ** RTKPOST only
Antenna Type	Select the type of the base-station antenna. To select the antenna type, set Receiver Antenna PCV File in Files. If "*" is used, antenna type and antenna delta are recognized by the antenna information of RINEX OBS header (RTKPOST) or RTCM antenna information (RTKNAVI).	<b>ant2-</b> <b>anttype</b>	
Delta-E/N/U	Set the delta position of the base-station antenna as E/N/U offsets of ARP position with refer to the marker (m).	<b>ant2-</b> <b>antdel1e,</b> <b>antdel1n,</b> <b>antdel1u</b>	

Item	Descriptions	Configuration File	Notes
Station Position File	<p>Input the station position file path to retrieve the position from the station list. The station position file is a text file which contains the multiple lines. Each line represents a record for a station. A record contains:</p> <ul style="list-style-type: none"> <li>- <b>Latitude (deg)</b></li> <li>- <b>Longitude (deg)</b></li> <li>- <b>Ellipsoidal height (m)</b></li> <li>- <b>Station ID</b></li> <li>- <b>Station name</b></li> </ul> <p>separated by spaces. The line starting "%" is treated as the comment line. An example station position file is found at <b>rtklib_&lt;ver&gt;\data\stations.pos</b>.</p> <p>SINEX station positions can be used as well. An sample SINEX file is found at  <b>rtklib_&lt;ver&gt;\data\igs10P1565_wcov.snx</b></p>	<b>file-staposfile</b>	

If you set Station Position File in "Positions" tab, you can select the position of the rover or the base-station antenna from the station list in "Stations" dialog by pushing  button. You can load Lat/Lon/Hgt format or SINEX format file. The file type is automatically recognized. You can search a station Id or Name with  button with a specified word in the Find field.



Figure 3.5-6 Positions Dialog of RTKNAVI and RTKPOST Options

(6) Files

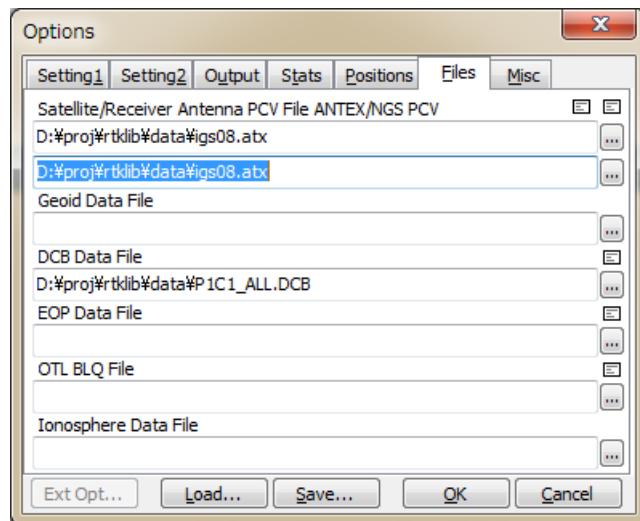


Figure 3.5-7 Options Dialog (Files) of RTKNAVI and RTKPOST

Item	Descriptions	Configuration File	Notes
Satellite Antenna PCV File ANTEX	If you use the precise ephemeris or SSR correction, input the ANTEX antenna parameters file path for the satellite antenna PCV (phase center variation) correction. Usually use latest <b>igs08.atx</b> file provided by IGS.  An example of the ANTEX file is found at <b>rtklib_&lt;ver&gt;\data\igs08.atx</b> .	<b>file-satantfile</b>	
Receiver Antenna PCV File ANTEX or NGS PCV	If you apply the receiver antenna phase center offset and PCV correction, input ANTEX or NGS type antenna parameters file path.  An example of the antenna parameter file is found at <b>rtklib_&lt;ver&gt;\data\igs08.atx</b> . or <b>rtklib_&lt;ver&gt;\data\ngs_abs.pcv</b> .	<b>file-rcvantfile</b>	
Geoid Data File	Input the file path of the geoid data file if selecting the external model as Geoid Model.	<b>file-geoidfile</b>	
DCB Data File	Input the file path of DCB correction for PPP in CODE format.  An example of the antenna parameter file is found at: <b>rtklib_&lt;ver&gt;\data\P1C1_ALL.DCB</b> <b>rtklib_&lt;ver&gt;\data\P2C2.DCB</b> <b>rtklib_&lt;ver&gt;\data\P1P2_ALL.DCB</b>	<b>file-dcbfile</b>	
EOP Data File	Input the file path of an EOP data file. The format of the EOP data file shall be IGS ERP format version. 2 [62].	<b>file-eopfile</b>	
Ocean Loading BLQ Format	Input the file path of an OTL coefficients file. The format of the OTL coefficients file is BLQ format.[63]	<b>file-blqfile</b>	
Ionosphere Data File	Input the file path of an ionospheric corrections file (*.yyi)	-	RTKPOST only
FTP/HTTP Local Directory	Input the local directory for FTP/HTTP download. The downloaded files are save in the directory.	<b>file-tempdir</b>	RTKNAVI only

## (7) Misc (RTKNAVI)

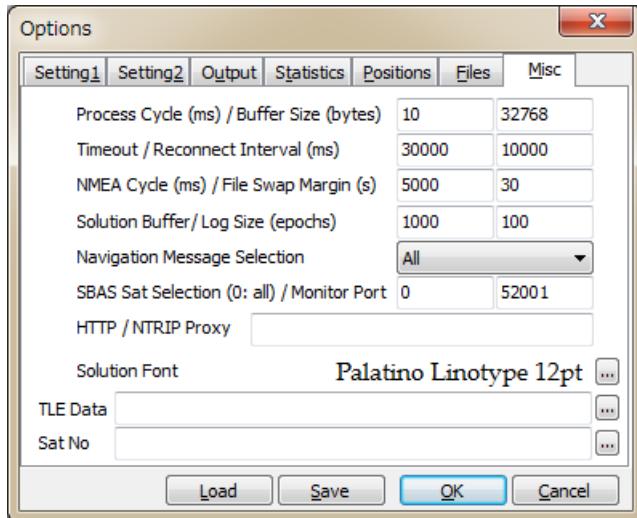


Figure 3.5-8 Options Dialog (Misc) of RTKNAVI

Item	Descriptions	Configuration File	Notes
Processing Cycle/ Buffer Size	Set the processing cycle time of in ms. Usually set 100 ms or less value.	<b>misc-svrcycle</b>	
Timeout/ Re-connect Interval	Set the timeout and re-connect interval for TCP client and NTRIP client connections in ms. If the timeout time expired without sever response, RTKNAVI retries to connect to server after waiting for the re-connect interval.	<b>misc-timeout, reconnect</b>	
NMEA Cycle/ File Swap Margin	Set the NMEA GPGGA transmission cycle to NRTK server in ms. If output or log file swap enabled, set the overlapped periods between the previous and the new output files in second. If you set it to 0, the periods of these files are not overlapped. This feature is to avoid the missing of transient data by the output file swapping.	<b>misc-nmeacycle</b>	
Solution Buffer/Log Size	Set the internal solution buffer size and log size in epochs. To increase the length of the receiver trajectory on "RTK Map", increase the solution buffer size.	<b>misc-buffsize</b>	
Navigation Message Selection	Select navigation messages to be used. - (1) All: In any of the input streams - (2) Rover: In the rover receiver stream - (3) Base Station: In the base station receiver stream - (4) Correction: In the correction stream	<b>misc-navmsgsel</b>	
SBAS Satellite Selection	If SBAS DGPS correction enabled, input SBAS satellite PRN number to be used. If you input 0, all available SBAS satellites are used.	<b>misc-sbasatsel</b>	Default: 52001

Item	Descriptions	Configuration File	Notes
Monitor Port	Set monitor port number. The monitor port is TCP server port to connect from outside or by RTKPLOT for real-time solution monitor. If multiple instances of RTKNAVI are generated, the following numbers are used automatically. If 0 is set, the monitor port is not used.	-	
HTTP/NTRIP Proxy	Set HTTP/NTRIP proxy server address and port number as <address>:<port> form to connect the NTRIP Caster via a HTTP-Proxy server.	<b>misc-proxyaddr</b>	
Solution Font	Select the font of the solution display in the main window.	-	
TLE Data	Specify NORAD TLE (two line element) satellite orbit element data file. TLE data are used to compute satellite positions for skyplot if the satellite ephemeris is unavailable. Both of two line format or three line format of TLE data can be used. A sample TLE data can be found at: <b>rtklib_&lt;ver&gt;/data/catalbe_21_2013_01_09_pm.txt.</b>	-	*1
Sat No	Specify the satellite number file which is to connect GNSS satellite/PRN numbers and TLE satellite catalog numbers in NORAD TLE data file. A sample satellite number file can be found at <b>rtklib_&lt;ver&gt;/data/TLE_GNSS_SATNO.txt.</b>	-	

\*1 To obtain the latest TLE data file, refer the following URLs:

(a) CelesTrack: <http://celestak.com>

(b) SpaceTrack: <http://www.space-track.org>

## (8) Misc (RTKPOST)

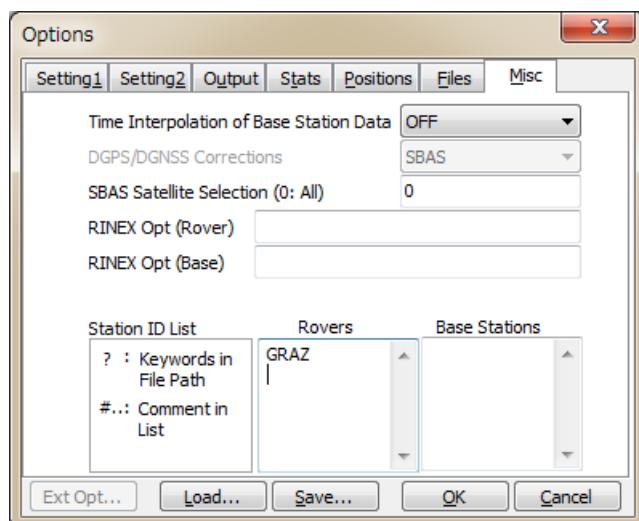


Figure 3.5-9 Options Dialog (Misc) of RTKPOST

Item	Descriptions	Configuration File	Notes
Time Interpolation of Base Station Observation Data	Select ON to enable time interpolation of base station data. If selecting ON, the base station data are linearly interpolated to the rover epoch and DD (double-difference) is made with them. If not checked, nearest epoch of base station data is used for DD.	<b>misc-timeinterp</b>	
SBAS Satellite Selection	If SBAS DGPS correction enabled, input SBAS satellite PRN number to be used. If you input 0, all available SBAS satellites are used.	<b>misc-sbasatsel</b>	
RINEX Opt (Rover)	Specify RINEX read options for rover RINEX observation data as follows. Multiple options can be input separated by spaces.  <pre>-GLss[=+n.nn]: select GPS signal ss -RLss[=+n.nn]: select GLO signal ss -ELss[=+n.nn]: select GAL signal ss -JLss[=+n.nn]: select QZS signal ss -CLss[=+n.nn]: select BDS signal ss -SLss[=+n.nn]: select SBS signal ss  ss      : signal id (refer Appendix D.3) =+n.nn: phase shift +n.nn (cycle) to be          add to carrier-phase observables</pre> <p>Without the option and multiple signals in a frequency, RTKLIB select a signal for a frequency according to the default priorities of signals. Refer Appendix D.4 Default Priorities of Multiple Signals.</p>	<b>misc-rnxopt1</b>	
RINEX Opt (Base)	Specify RINEX read options for base-station RINEX observation data same as the RINEX Opt (Rover).	<b>misc-rnxopt2</b>	
Station ID List	For batch processing with multiple input files or multiple sessions, you can set input file paths or output file path containing the following keywords:  <code>%y, %Y, %m, %d, %n, %w, %D, %h, %H, %r, %b</code>  The keywords are replaced by the proper values or expanded for multiple session analysis. To enable time keywords, set Time Start, Time End and Unit (optional) in the main window. For keywords <code>%r, %b</code> , input Rover List or Base Station List below. For online reference, push "?" button.		
Rovers	Input the rover ID list to replace keyword <code>%r</code> in input and output file paths. The line starting with "#" is treated as a comment.	-	
Base Stations	Input the base station ID list to replace keyword <code>%b</code> in input and output file paths. The line starting with "#" is treated as a comment.	-	

### 3.6 Convert Receiver Raw Data to RINEX with RTKCONV

RINEX (Receiver Independent Exchange Format) is a standard GPS/GNSS data format supported by many receivers or GPS/GNSS post-processing analysis software. RTKLIB post-processing analysis AP RTKPOST can also handle RINEX data files as inputs. For preparing RINEX files, RTKLIB provides the converter AP RTKCONV, which translates receiver raw, RTCM and BINEX messages to RINEX OBS (observation data), RINEX NAV (GNSS navigation messages). RTKCONV can also extract SBAS messages from the receiver raw data and output the SBAS log file.

The supported RINEX versions are 2.10, 2.11, 2.12, 3.00, 3.01 and 3.02 (draft) with RTKLIB extensions. Refer Appendix B.2 for SBAS log files and Appendix D.1 for supported RINEX files.

- (1) Execute the binary AP file `<install dir>\rtklib_<ver>\bin\rtkconv.exe`. You can see the main window of RTKCONV.

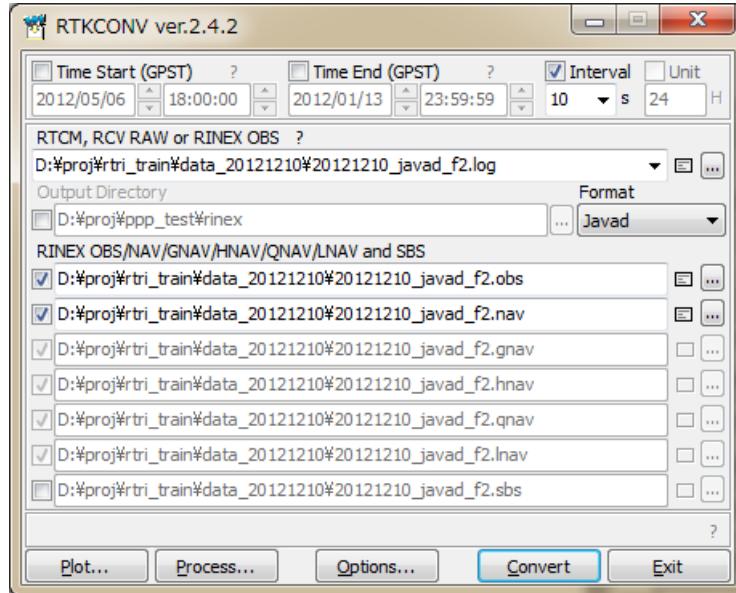


Figure 3.6-1 Main Window of RTKCONV

- (2) Input the receiver raw data file path to the text field RTCM, RCV RAW or RINEX OBS. Fill in the file path directly or select the file with the file selection dialog by pushing  button. You can also drag and drop the icon of the raw data file to the main window of RTKCONV. Supported receiver raw data formats are as follows. Refer Appendix D.2 for detailed supported messages for each format. Refer

Appendix D.3 for the relationship to be used for conversion between multiple observation codes expressed in RINEX version 2, RINEX version 3, RTCM 3 MSM and BINEX.

(a) RTCM2	: RTCM 2.3
(b) RTCM3	: RTCM 3.0, 3.1 (with amendment 1-5) and 3.2
(c) NovAtel OEM6	: NovAtel OEM4/OEMV/OEM6 and OEMStar binary format
(d) NovAtel OEM3	: NovAtel OEM3 (Millennium) binary format
(e) u-blox	: u-blox LEA-4T, 5T, 6T binary format
(f) Superstar II	: NovAtel Superstar II binary format
(g) Hemisphere	: Hemisphere Crescent/Eclipse binary format
(h) SkyTraq	: SkyTraq S1315F binary format
(i) GW10	: Furuno GW-10-II/III binary format
(j) Javad	: JAVAD GRIL/GREIS binary format
(k) NVS BINR	: NVS NV08C BINR binary format
(l) BINEX	: BINEX format (only big-endian, forward, regular CRC messages)
(m) RINEX	: RINEX OBS/NAV format.

If the wild-card (\*) is used in the file path, the wild-card is expanded to multiple files and they will be read. By setting "RINEX" as the input file format and setting output RINEX version, you can convert RINEX version 2 to version 3 or RINEX version 3 to version 2. In this case, you can extract and output RINEX data with selected satellites, signals, time span or time interval of input RINEX data.

- (3) Select the format with the pull down menu Format. If you select "Auto", RTKCONV recognizes the file format according to the following file extensions.

(a) RTCM2	: . <b>rtcm2</b>
(b) RTCM3	: . <b>rtcm3</b>
(c) NovAtel OEM6	: . <b>gps</b>
(d) u-blox	: . <b>ubx</b>
(e) Superstar II	: . <b>log</b>
(f) Hemisphere	: . <b>bin</b>
(g) SkyTraq	: . <b>stq</b>
(h) Javad	: . <b>jps</b>
(i) BINEX	: . <b>bnx, binex</b>
(j) RINEX	: . <b>obs, *o, nav, *n, *p, *g, *h, *q, *l</b>

- (4) Input the output paths of RINEX OBS (observation data), RINEX NAV (GPS navigation messages), RINEX GNAV (GLONASS navigation messages), RINEX HNAV (GEO satellite navigation messages), RINEX QNAV (QZSS navigation messages), RINEX LNAV (Galileo navigation messages) and SBAS Log files. Fill in the file path directly or select the file with the file selection dialog by pushing  button. If you do not check the checkbox left, the file is not output. RINEX GNAV, HNAV, QNAV and LNAV are supported only in RINEX version 2. If selecting version 3 as the output RINEX, all navigation data are output to a combined (mixed) NAV type RINEX file. The output files can include keywords. The keywords are replaced by time, date or station ID. Push "?" button, to refer the keyword replacements in output files.
- (5) If you want to output the files to a directory, check "Output Directory" and input the output directory. The directory can be selected by pushing  button. If the "Output Directory" is not checked, the files are output to the same directory of the input file. If the output directory is not exist, the directory is automatically created. Even in this case, the parent directory always must be exist. Otherwise, the directory creation is failed.
- (6) You can set the start time or end time optionally by checking and setting the "Time Start (GPST)" or "Time End (GPST)" field upper in the main window. You can also set the time interval option by checking and setting the field "Interval". If you input "Time Start" and "Time End", check "Unit" and input "Time Unit", you can initiate multiple-session conversion. In this case, please include keywords to be replaced by date and time in the input file path and the output file paths. Note that RTKCONV does not confirm the overwrite if the output file exists in such a multi-session conversion case.
- (7) You can push  button to configure RINEX options. Set the options for RINEX headers, navigation systems, observation types or frequencies with "Options" dialog. If you check RINEX Name, the output file paths are compliant to the RINEX file name convention. Please input Station ID to complete the output file names for RINEX file name convention. The "Option" field is receiver-dependent options. Specify the options separated by spaces like "**-EPHALL -GL1X**". For details, refer Appendix D.5 Receiver Dependent Input Option. For RINEX 3, you had better check "Scan Obs Types" to obtain effective OBS TYPES in input files. In this case, the input files are scanned to obtain available OBS TYPE list as the first conversion path and then RTKCONV outputs RINEX as the second conversion path. If "Scan Obs Types" unchecked, the OBS TYPES in output RINEX files are determined by the default OBS TYPES set depending on the input format and the "Signal Mask" settings described below.

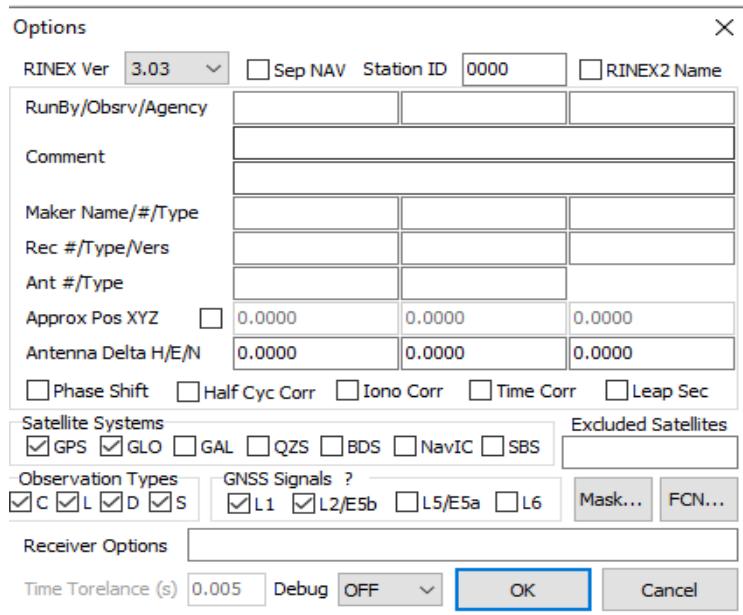


Figure 3.6-2 Options Dialog of RTKCONV

- (9) To set the mask of OBS TYPES to output to the observation file, use the "Signal Mask" dialog shown by pushing **Mask...** button in the "Options" dialog. You shall check signals to be output to RINEX observation file in the dialog. You shall uncheck signals not to output to the RINEX observation file. Note that the signals not in the default OBS TYPES set or not in input data are not output even if checked. Use **Set All** or **Unset All** button to select or deselect all signals. Refer Appendix D.3 for observation types or signal IDs in the input data and the output RINEX files.

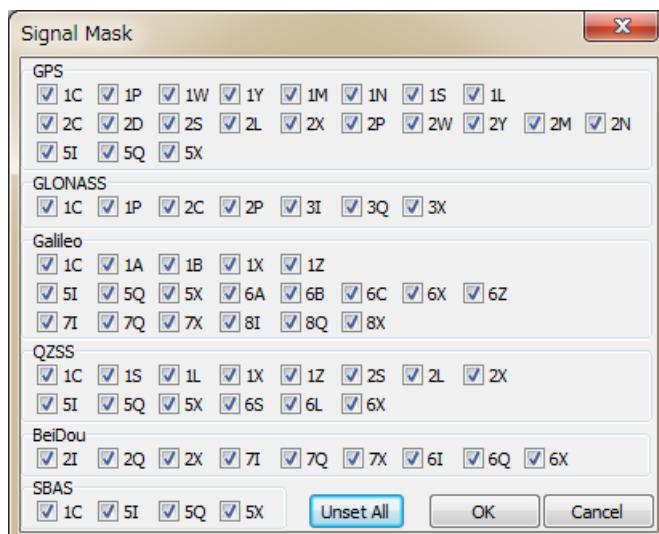


Figure 3.6-3 Signal Mask Dialog of RTKCONV Options

- (8) Push **Convert** button to start converting the receiver raw data to RINEX and SBAS log files. If you

want to stop the conversion on the way, push **Abort** button. The status is displayed in the message area lower center in the main window. The message **O=nnn** means the number of converted observation data (epochs). The message **N=nnn**, **G=nnn**, **H=nnn**, **Q=nnn**, **L=nnn**, **S=nnn** and **E=nnn** means the number of navigation messages (NAV, GNAV, HNAV, QNAV and LNAV), SBAS messages and errors, respectively.

- (14) After finishing the conversion, you can see the observation data plot by pushing **Plot...** button with RTKPLOT. Refer 3.7 Plot and View Solutions and Observation Data for details. You can also view the output file with Text Viewer by pushing **File** button.
- (15) By pushing **Process...** button, you can execute RTKPOST\_MKL AP to process the converted RINEX OBS/NAV files. Refer 3.4 Post-processing Analysis with RTKPOST for details.

### 3.7 View and Plot Solutions with RTKPLOT

RTKLIB contains the AP RTKPLOT to view and plot the positioning solutions by RTKPOST and RTKNAVI with graphical user interface. RTKPLOT also accepts general NMEA 0183 files or streams to generate the solution plot.

- (1) Execute the binary AP file <install dir>\rtklib\_<ver>\bin\rtkpplot.exe. You can see the main window of RTKPLOT. By pushing [Plot...] button or some buttons of RTKPOST and RTKCONV, RTKPLOT is also executed.

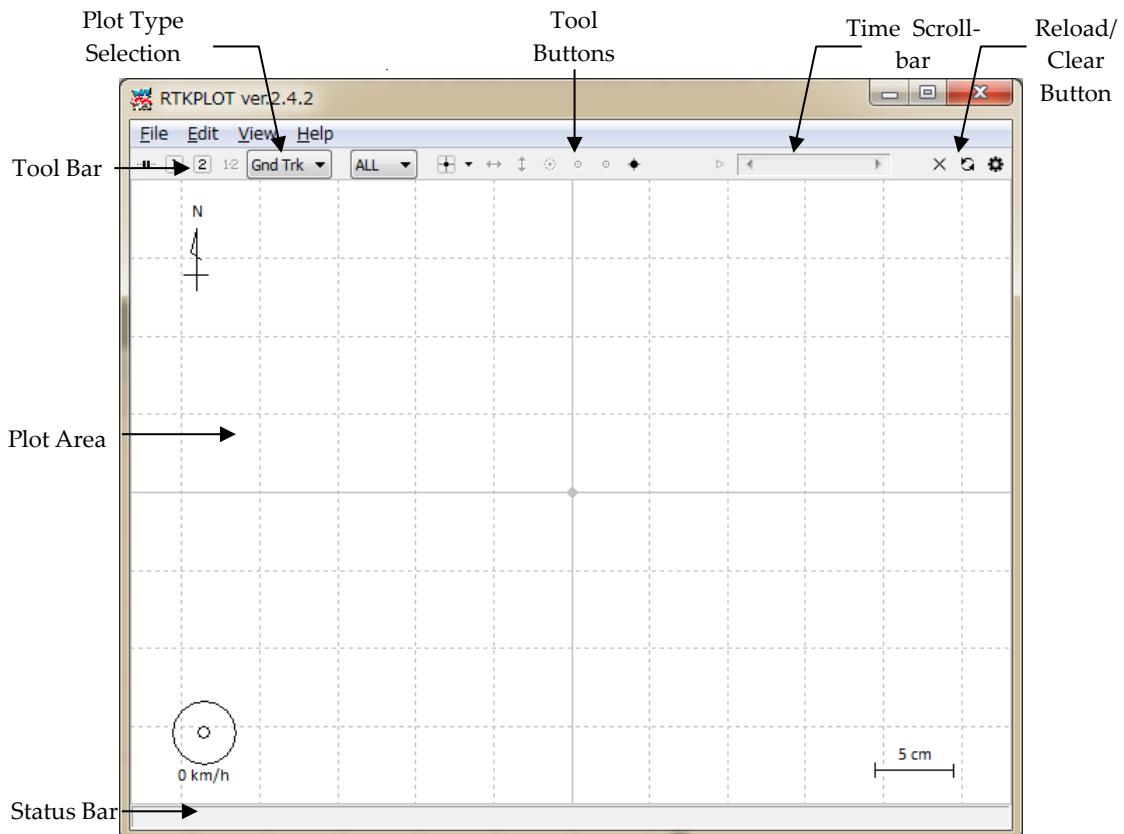


Figure 3.7-1 Main Window of RTKPLOT

- (2) Execute the menu "File" - "Open Solution 1" and select the solution file with the file selection dialog. The input solution file can be RTKLIB solution format or NMEA-0183. If the file format is NMEA-0183, the file must contain at least NMEA GPRMC and GPGGA sentences. If the solution file is valid, the receiver ground track is plot in the window on the map. The color of the marks, lines and grid in the plot can be changed with the menu "Edit" - "Options". The status bar at the bottom of the main window

also shows the time range, the number of solution epochs (**N=nnnn**), the baseline length (**B=0 . 0-x . xkm**), the number and percentage of each quality solutions (**Q=1:nnn (pp%) , 2:nnn (pp%) , . . .**). The quality flag Q and the marker color means: 1: Fixed, 2: Float, 4: DGPS, 5: Single (the colors are changeable with the plot options). To screen the marks by the quality flag Q, select the second pull down menu in the Tool Bar. By drag and drop of the solution file icon to the main window of RTKPLOT, you can also read and plot the solution file.

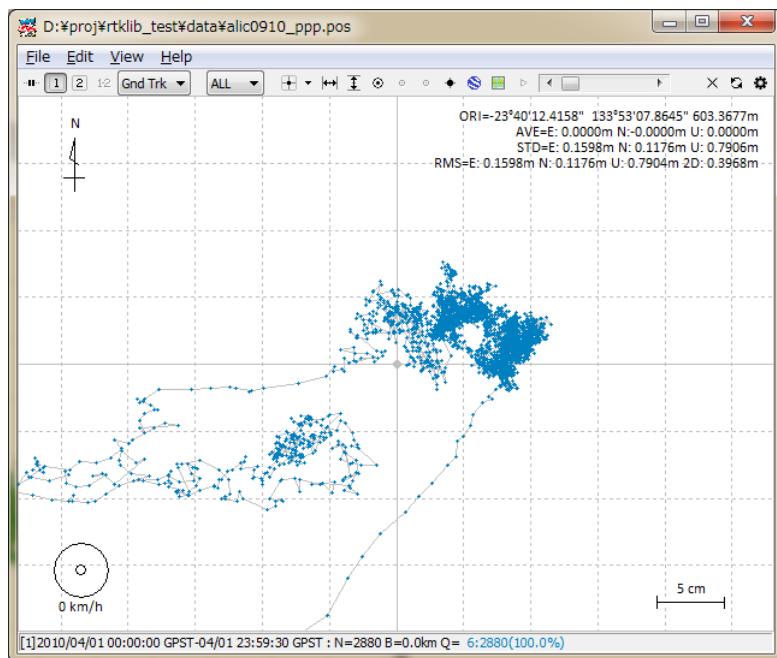


Figure 3.7-2 GND TRK Plot by RTKPLOT

- (3) By dragging the mouse with the left button down on the plot, you can drag the map up, down, left and right . You also change the scale of the map by dragging the mouse up or down with the right button or by rotating the scroll wheel of the mouse.
- (4) By selecting the plot type pull down menu right in Tool Bar, you can switch the plot to E/N/U components of receiver position (Position), E/N/U components of receiver velocity (Velocity) or E/N/U components of receiver acceleration (Accel). You can drag the X/Y-axis with left-button-dragging and change the scale with right-button-dragging at the X/Y-axis area. You can hide or show the three plots by pushing three buttons right of the plot type pull down menu.

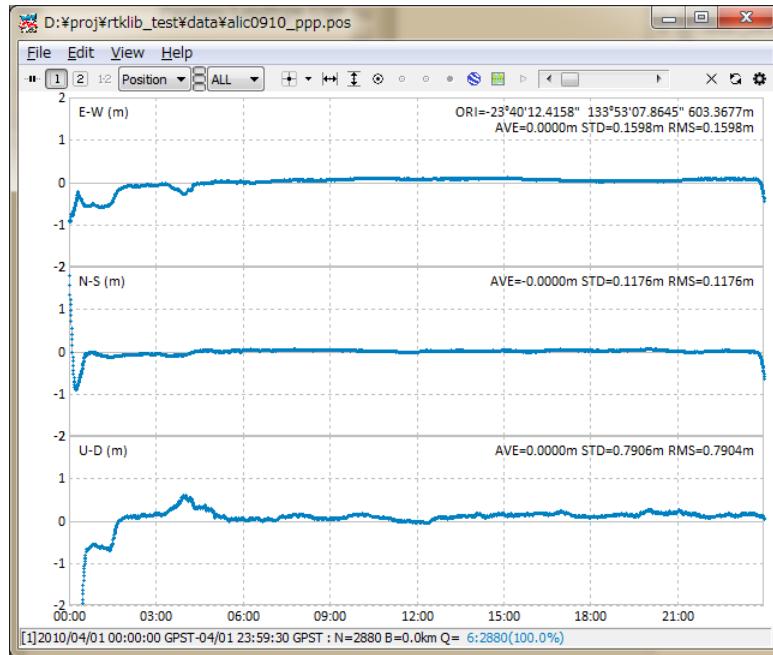


Figure 3.7-3 POSITION Plot by RTKPLT

- (5) By selecting the plot type pull down menu, you can switch the plot to NSat/Age/Ratio (number of valid satellites, age of differential, ratio factor of ambiguity validation). If you set the "Output Solution Status" option to "Residuals", you can show a residuals plot. You can switch the frequency by selecting L1/LC, L2 or L5. In the residuals plot mode, you can select a satellite with right pull down menu as well as all satellites. In the residuals plot of carrier-phases, the red lines indicate cycle slips and gray lines indicate parity unknown flags (That mean the half-cycle ambiguities in carrier-phase are not resolved).

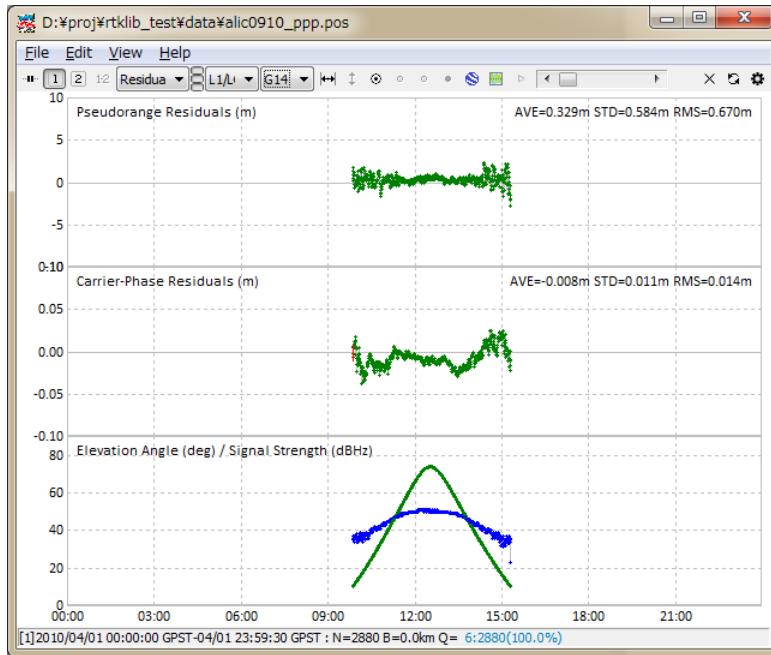


Figure 3.7-4 RESIDUALS Plot by RTKPLOT

- (6) By pushing tool buttons in the Tool Bar, you can center the current position with , adjust the scale of X-axis with , adjust the scale of Y-axis with , display the current position as a large mark with , fix the current track position at the horizontal center with , fix the current track position at the vertical center with , start animation with and stop animation with . You can also slide "Time Scroll Bar" to change the current epoch. To clear read data, execute the menu "File" - "Clear" or push button in the Tool Bar. To reload the solution file(s), execute the menu "File" - "Reload" or push button in the Tool Bar.
  
- (7) By executing the menu "File" - "Open Map Image", you can read a JPEG image and draw the map image on the background of the plot in case of the "Gnd Trk" plot type. The image can be enabled or disabled by pushing button in the Tool Bar.

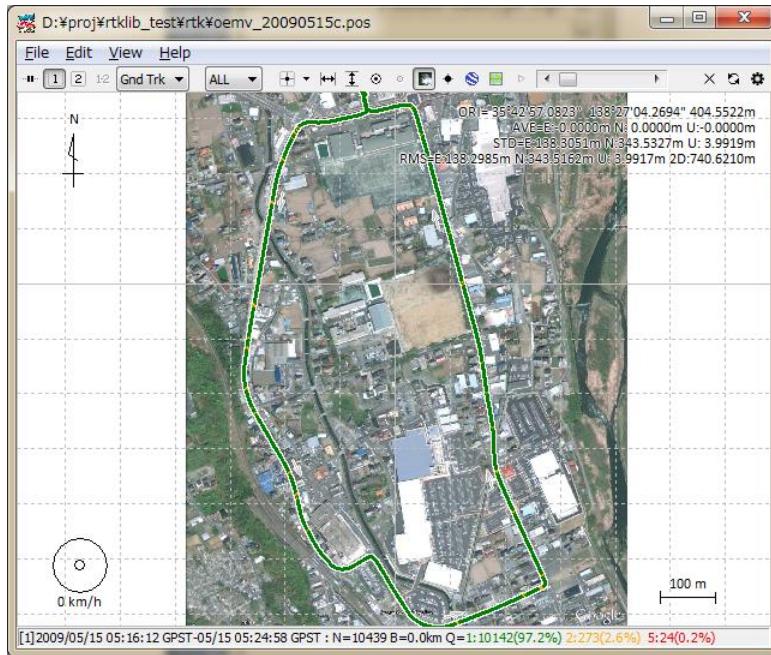


Figure 3.7-5 Map Image Overlay by RTKPLOT

- (8) To adjust the position in the map image, execute the menu "Edit" - "Map Image" and input latitude and longitude of the image-center, image-scale along X or Y-axis in "Map Image" dialog. If you finish it, push **Save Tag** button to save the adjustment information to an image-tag-file. The path of the image-tag-file is the original map image file path + **.tag**. If the image-tag-file already exists, it is automatically read with the map image itself. The current version does not support rotation of the map image. Please select the map image in which the north direction is properly aligned to the upper direction. For example, you can get a JPEG image by the menu "File" - "Save" of Google Earth. To fix north to upper, push "N" button in Google Earth. To avoid the distortion of the map image, set the coordinates origin inside of or near the map image.

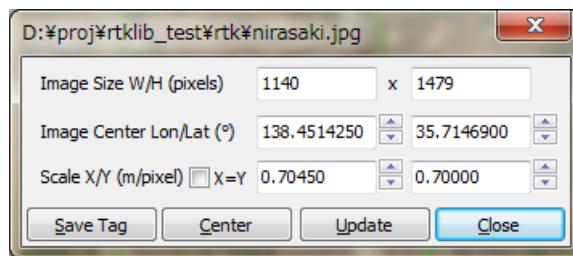


Figure 3.7-6 Map Image Options Dialog of RTKPLOT

- (9) In version 2.4.2, Google Map View for RTKPLOT are added to plot the solution on Google Map. To open Google Map View, execute the menu View - Google Map View of RTKPLOT after the solution

read. The toolbar buttons  or  can also be used to show these views. Notes that the Internet connection is always necessary to use these views based on services provided by Google.

- (10) In the Google Map View, the only a toolbar button to fix the track point center as same as the Google Earth View. Other operations for the Google Map View, use the controls in the Google Map View.



Figure 3.7-8 Google Map View of RTKLIB

- (11) By executing the menu "Edit" - "Waypoints...", you can see the "Waypoints" dialog. With the dialog, you can load, save, add and delete the way-points as the list form. By pushing **Add** button and editing the point name, the current receiver position can be added to the way-point list. The positions of the waypoints are shown on the "Gnd Trk" plot when  button down.

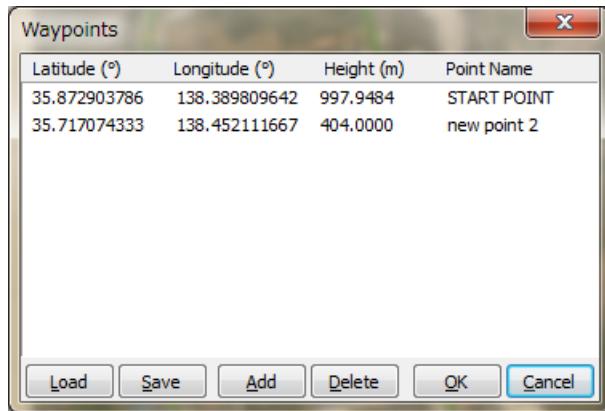


Figure 3.7-9 Waypoints Dialog of RTKLIB

## RTKLIB Demo5 Manual

- (12) To plot multiple solution file, execute the menu "File" - "Open Solutions-2" and select file with the file selection dialog. You can switch the plot on/off of the solution 1 and 2 with [1] [2] buttons in Tool Bar. To Plot the difference of the solution 1 and the solution 2, push [1-2] button in Tool Bar.
- (13) To set the time range and time interval of the solutions, execute the menu "Edit" - "Time Span/Interval" and check and set the Time Start, Time End and Interval field in "Time Span/Interval" dialog.

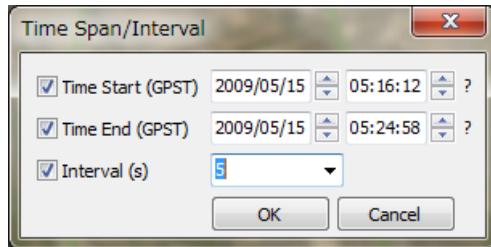


Figure 3.7-10 Time Span/Interval Dialog of RTKPLOT

- (14) By Executing the menu "Edit" - "Solution Source", you can view the source of solutions as the text form.

D:\proj\rtklib_test\rtk\oemv_20090515c.pos							
	Find	Option... Close					
% GPST	latitude(deg)	longitude(deg)	height(m)	Q	ns	sdp(m)	sde(m)
2009/05/15 05:16:12.100	35.718950852	138.450589137	404.8966	5	11	2.8485	2.3232
2009/05/15 05:16:12.150	35.718950841	138.450589143	404.9019	5	11	2.8485	2.3232
2009/05/15 05:16:12.200	35.718950840	138.450589177	404.8965	5	11	2.8485	2.3232
2009/05/15 05:16:12.250	35.718950857	138.450589145	404.9099	5	11	2.8485	2.3232
2009/05/15 05:16:12.300	35.718950884	138.450589172	404.9032	5	11	2.8485	2.3232
2009/05/15 05:16:12.350	35.718950852	138.450589196	404.9073	5	11	2.8485	2.3232
2009/05/15 05:16:12.400	35.718950869	138.450589161	404.9116	5	11	2.8485	2.3232
2009/05/15 05:16:12.450	35.718950845	138.450589159	404.9039	5	11	2.8485	2.3232
2009/05/15 05:16:12.500	35.718950869	138.450589146	404.9106	5	11	2.8485	2.3232
2009/05/15 05:16:12.550	35.718950839	138.450589213	404.9129	5	11	2.8485	2.3232
2009/05/15 05:16:12.600	35.718950864	138.450589170	404.9060	5	11	2.8485	2.3232
2009/05/15 05:16:12.650	35.718950868	138.450589193	404.9066	5	11	2.8485	2.3232
2009/05/15 05:16:12.700	35.718950863	138.450589171	404.9045	5	11	2.8485	2.3232
2009/05/15 05:16:12.750	35.718950863	138.450589151	404.9084	5	11	2.8485	2.3232
2009/05/15 05:16:12.800	35.718950842	138.450589176	404.9086	5	11	2.8485	2.3232
2009/05/15 05:16:12.850	35.718950875	138.450589195	404.9031	5	11	2.8485	2.3232
2009/05/15 05:16:12.900	35.718950877	138.450589194	404.9042	5	11	2.8485	2.3232
2009/05/15 05:16:12.950	35.718950846	138.450589164	404.9064	5	11	2.8485	2.3232
2009/05/15 05:16:13.000	35.718950847	138.450589180	404.9109	5	11	2.8485	2.3231
2009/05/15 05:16:13.050	35.718950854	138.450589182	404.9029	5	11	2.8485	2.3231
2009/05/15 05:16:13.100	35.718950861	138.450589219	404.9099	5	11	2.8485	2.3231
2009/05/15 05:16:13.150	35.718950853	138.450589162	404.9042	5	11	2.8485	2.3231
2009/05/15 05:16:13.200	35.718950882	138.450589177	404.9019	5	11	2.8485	2.3231
2009/05/15 05:16:13.250	35.718950864	138.450589172	404.9168	5	11	2.8485	2.3231
2009/05/15 05:16:13.300	35.718955206	138.450598594	409.0196	2	8	0.8737	0.8872
2009/05/15 05:16:13.350	35.718955216	138.450598616	409.0244	2	8	0.6265	0.6361
2009/05/15 05:16:13.400	35.718955212	138.450598623	409.0235	2	8	0.5140	0.5218
2009/05/15 05:16:13.450	35.718955218	138.450598624	409.0239	2	8	0.4462	0.4530

Figure 3.7-11 Solution Source View of RTKPLOT

- (15) To plot solutions in real-time, execute the menu "File"- "Connection Settings" and set solution

parameters in "Connection Setting" dialog. You can select Stream Type, Stream Option (Opt), Stream Commands (Cmd), Solution Format, Time Format, Lat/Lon Format and Field Sep for both of solution-1 and solution-2. After setting the connection parameters, execute the menu "File"- "Connect" or push  button in Tool Bar. To disconnect the external device, execute menu "File"- "Disconnect" or push connect button again. For example, if selecting serial as Stream Type and NMEA0183 as Solution Format, you can monitor external receiver's NMEA output in the RTKPLOT window.

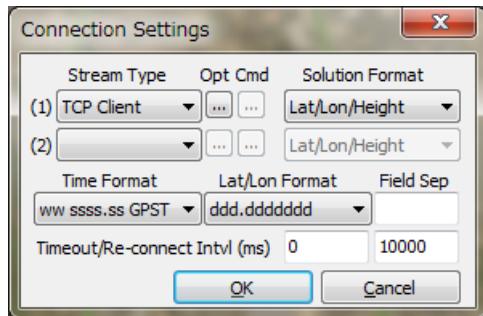


Figure 3.7-12 Connection Settings Dialog of RTKPLOT

- (16) By pushing  button of RTKNAVI, RTKPLOT automatically runs and connects to the monitor port of RTKNAVI. If you want to connect to RTKNAVI executed on the remote PC, configure the connection setting in the Connection Setting dialog like TCP Client as Stream Type, IP address of the remote PC and port number of monitor port of RTKNAVI and connect to the remote RTKNAVI. In this case, multiple client connections are allowed from multiple PCs running RTLPLOTs.
- (17) To configure the plotting options for RTKPLOT, execute the menu "Edit" - "Options..." and set the options with the following "Options" dialog.

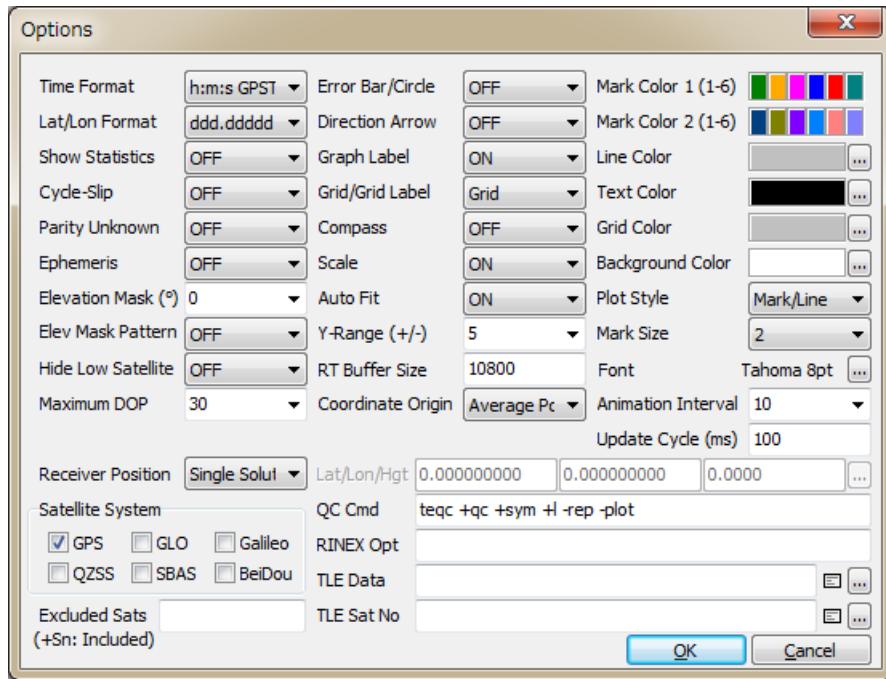


Figure 3.7-13 Options Dialog of RTKPLOT

Item	Descriptions	Notes
Time Format	Select time format. (www/sss, h:m:s GPST, h:m:s UTC or h:m:s LT).	
Lat/Lon Format	Select latitude and longitude format. (ddd.ddddd or ddd mm ss.ss)	
Show Statistics	Set whether statistics are shown or not.	
Cycle-Slip	Set whether cycle-slip position is shown or not in the satellite visibility plot. If "LG Jump" selected, dual frequency geometry free LC (linear combination) is used to detect cycle-slips. In the case of "LLI Flag", LLI (loss-off-lock indicator) in RINEX observation data is used. Cycle-slips are shown as red vertical lines in the Satellite Visibility plot.	
Parity Unknown	Set whether parity unknown status is shown or not in the satellite visibility plot. Parity unknown epochs are shown as gray vertical lines in the satellite visibility plot.	
Ephemeris	Set whether ephemeris status is shown or not in the satellite visibility plot. Ephemerides are shown as the grey line under the observation data. Gray dots mean the Toe (time of ephemeris). Red ephemeris lines mean the satellite unhealthy.	
Elevation Mask	Set the elevation mask angle (deg) for the satellite visibility plot. The elevation mask is also used for DOP/NSat plot.	
Elevation Mask Pattern	Set whether elevation mask pattern is used or not.	
Hide Low Satellite	Set whether low elevation satellites under the elevation mask and the elevation mask pattern are shown or not.	

Item	Descriptions	Notes
Maximum DOP	Set the y-axis limit of DOP/NSat plot.	
Receiver Position	Set the receiver position for the satellite visibility plot or skyplot. "Single Solution" uses single-point results as receiver positions by using observation and navigation data. For moving receivers, you shall set it. "Lat/Lon/Hgt" uses latitude, longitude and height for static receivers specified as the following Lat/Lon/Hgt fields. "RINEX Header" uses "APPROX POSITION XYZ" in RINEX observation data header as the receiver position.	
Satellite System	Check selected navigation systems for plots.	
Excluded Sats	Set excluded satellites. Fill in the satellite number or ID separated by spaces.	
Error Bar/Circle	Set whether error bar or error circle is shown or not in solution display. You can select "Bar/Circle" or "Dots" as the format.	
Direction Arrow	Set whether direction arrow and velocity arrow are shown or not in the solution ground track plot.	
Graph Label	Set whether graph labels are shown or not in solution display.	
Grid/Grid Label	Set whether grid and grid labels are shown or not in solution display. Set it to "Circles" or "Circles/Label" for circular grids.	
Compass	Set whether compass is shown or not in the solution ground track plot.	
Scale	Set whether scale is shown or not in the solution ground track plot.	
Auto Fit	Set whether scale is adjusted automatically or not for solution plots.	
Y-Range (+/-)	Set the range of Y-axis in solution plots.	
RT Buffer Size	Set the buffer size for real-time solution plots in epochs. The older solutions over the buffer size are deleted for real-time plots.	
Coordinate Origin	<p>Select the origin position of solution display as follows.</p> <ul style="list-style-type: none"> <li>- Start Pos: the first solution position</li> <li>- End Pos: the last solution position</li> <li>- Average Pos: the average of all solution positions</li> <li>- Linear Fit Pos: based on the linearly fitted positions</li> <li>- Base Station: the base station position.</li> <li>- Lat/Lon/Hgt: latitude, longitude and height specified</li> <li>- Auto Input: auto position as follows</li> <li>- Waypoint 1-10: a waypoint position</li> </ul> <p>If you select "Lat/Lon/Height", you have to input latitude, longitude and ellipsoidal height in the text fields below for the origin. If selecting "Auto Input", receiver ID is assumed as the 4-charactors of the solution file name head and the position is read from the position file. The position file can be selected by pushing ... button and pushing "Load" button in the position list dialog.</p>	

Item	Descriptions	Notes
Mark Color 1 (1-6)	Set the mark colors for solution No. 1 or observation data in plots. Click color panel right and select color with color selection dialog.	
Mark Color 2 (1-6)	Set the mark colors for solution No. 2 in plots.	
Line Color	Set the line color in plots.	
Text Color	Set the text color in plots.	
Grid Color	Set the grid color in plots.	
Background Color	Set the background color in plots.	
Plot Style	Select the plot style in plots. To delete all of markers and lines except for the track point mark, set it "None".	
Mark Size	Set the mark size in plots.	
Font	Select the font in plots. Push ... button and select the font with the font selection dialog.	
Animation Interval	Set the animation interval for solution or observation data plots.	
Update Cycle (ms)	Set the plot update cycle time in ms for real-time plots	
Lat/Lon/Hgt	Set latitude, longitude and height for the origin. Fill in the values directly or push ... button and select a station position.	
QC Command	Set QC command and its options for observation data. As default, TEQC QC mode options are set. The command is used for executing the menu "Edit" - "Obs Data QC...". The command must be in the command search path or in the directory for the RTKLIB executable files.	
RINEX Opt	Set RINEX read options. For RINEX read options, refer <code>&lt;install dir&gt;\rtklib_&lt;ver&gt;\src\rinex.c</code> for details.	
TLE Data	Specify NORAD TLE satellite orbit element data file. The TLE data are used to compute satellite positions for the skyplot if the satellite ephemeris unavailable. Both of two line format or three line format of TLE data can be used. A sample TLE data can be found at: <code>&lt;iinstall dir&gt;\rtklib_&lt;ver&gt;\data\catalbe_21_2013_01_09_pm.txt</code> .	See also RTKNAVI Options dialog
Sat No	Specify the satellite number file which is to connect GNSS satellite/PRN numbers and TLE satellite catalog numbers in NORAD TLE data file. A sample satellite number file can be found at <code>&lt;install dir&gt;\rtklib_&lt;ver&gt;\data\TLE_GNSS_SATNO.txt</code> .	

- (19) The following menus are provided in RTKPLOT. Some menus can be executed by pushing button on the Tool Bar.

Menu	Tool Bar	Description	Notes
Open Solution-1...		Open solution data No. 1.	* double click
Open Solution-2...		Open solution data No. 2.	* double click
Open Map Image...	-	Open map image data for the solution plot.	
Open Map Path...	-	Open map path data for the solution plot.	
Browse Solution	-	Show "Solution Browser" window and browse solution data.	
Open Obs Data...	-	Open observation data. Navigation data are also open automatically.	
Open Nav Data...	-	Open navigation data manually.	
Open Elev Mask...	-	Open elevation mask data.	
Visibility Analysis...	-	Execute satellite visibility analysis.	
File	Save Image...	-	Save the image of the plot to an image file.
	Save # of Sats/DOP...	-	Save the number of satellites and DOP to an text file.
	Save SNR,MP and AZ/EL..	-	Save the SNR, multipath and azimuth/elevation data to an text file.
	Connect...		Connect to external real-time solution streams.
	Disconnect...		Disconnect the connection.
	Connection Settings...	-	Show the "Connection Setting" dialog to configure the connection options.
	Reload		Reload solution, observation and navigation data.
	Clear		Clear solution, observation and navigation data.
	Exit	-	Close and exit RTKPLOT.
	Time Span/Interval...	-	Set the "Time Span/Interval" dialog to set time span and time interval.
Edit	Map Image...	-	Show the "Map Image" dialog to configure the size, position and scale of the image data.
	Waypoints...	-	Show the "Waypoint" dialog to add or modify the waypoints.
	Solution Source...	-	Show the source of the solution data by Text Viewer.
	Obs Data Source...	-	Show the source of the observation data by Text Viewer.
	Obs Data QC...	-	Execute QC of the observation data and show the results by Text Viewer.
	Copy To Clipboard	-	Copy the image of the plot to Clipboard.

Menu	Tool Bar	Description	Notes
Options		Show the "Options" dialog to configure plot options.	
Show Tool Bar	-	Show or hide the Tool Bar.	
Show Status Bar	-	Show or hide the Status Bar.	
Google Earth View...		Show Google Earth View.	
Google Map View...		Show Google Map View.	
Input Monitor 1...	-	Show the "Input Monitor" window for real-time input stream No.1.	
Input Monitor 2...	-	Show the "Input Monitor" window for real-time input stream No.2.	
Center Origin		Move the coordinates origin to the center of the plot.	
Fit Horizontal		Fit the horizontal range of the solution or observation data in the plot.	
View Fit Vertical		Fit the vertical range of the solution data in the plot.	
Show Track Point		Show or hide the track point of the solution or observation data in the plot.	
Fix Track Center		Fix the track point on the center of the plot.	
Fix Track Horizontal		Fix the track point horizontally on the plot.	
Fix Track Vertical		Fix the track point vertically on the plot	
Show Map Image		Show or hide the map image.	
Show Path/Waypoints		Show or hide the map path data.	
Animation Start		Start the animation of the plot.	
Animation Stop		Stop the animation of the plot.	
Help About...	-	Show the "About..." dialog.	

### 3.8 View and Plot Observation Data with RTKPLOT

RTKPLOT is also used to view and plot the RINEX observation data as well as satellite visibility analysis.

- (1) To plot the RINEX observation data, execute menu "File" -- "Open Obs Data" of RTKPLOT and select the RINEX observation and navigation message files. You can select multiple RINEX files. If you do not select the RINEX navigation message file, RTKPOST reads the observation data file paths with the extension replaced by **.\*nav (.obs)** or **.yyN, .yyG, .yyH, .yyQ, .yyL, .yyP (.yyO)** as the GPS, GLONASS, SBAS, QZSS, Galileo or combined navigation data. If you want read the RINEX navigation message file separately, execute the menu "File" - "Open Nav Messages". If the input files can be read properly, you can see a satellite visibility plot. You can use left-button or right-button drag to change the time span. You can also use some tool bar buttons as well as for the solution plot.

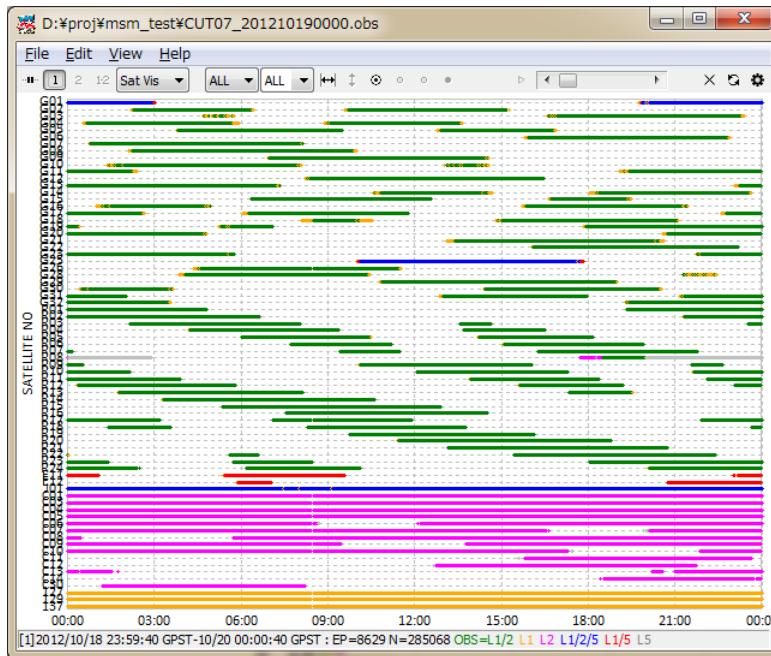


Figure 3.8-1 Satellite Visibility Plot of RTKPLOT

- (20) By selecting the pull down menu in Tool Bar, you can switch the plot to the satellite visibility in skyplot (SkyPlot) or the plot of the number of visible satellites, the DOP/NSat plot (DOP/NSat), the SNR/Multipath/Elevation plot (SNR/MP/EL) and the SNR/Multipath - EL plot (SNR/MP-EL). To show the SNR plots, the observation data must contain SNR (C/N0) information. To show the proper multipath plots, the observation data must contain dual-frequency (L1-L2 for GPS, GLONASS and QZSS, L1-L5 for Galileo, L2-L7 for BeiDou) pseudorange and carrier-phase. RTKPLOT internally

generates MP LC (linear combination) with these observables to plot the multipath. To compute the elevation angle of satellites, RTKPLOT also need satellite positions. In case of RINEX NAV files input, the satellite positions are computed by using satellite ephemerides in RINEX NAV files. If the RINEX NAV files are not available, RTKPLOT alternatively uses NORAD TLE (two line element) data set for satellite positions. In this case, you must specify the receiver position as the latitude, longitude and height form in the "Options" dialog. You can select signals and satellites with upper pull-down menus on the tool bar. In the satellite selection, "G", "R", "E", "J", "C" and "S" mean GPS, GLONASS, Galileo, QZSS, BeiDou and SBAS, respectively.

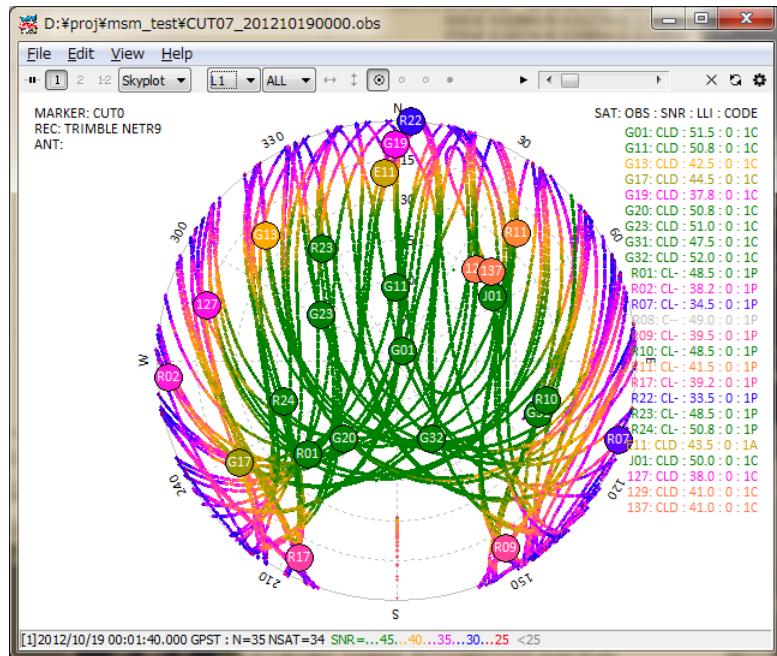


Figure 3.8-2 Skyplot by RTKPLOT

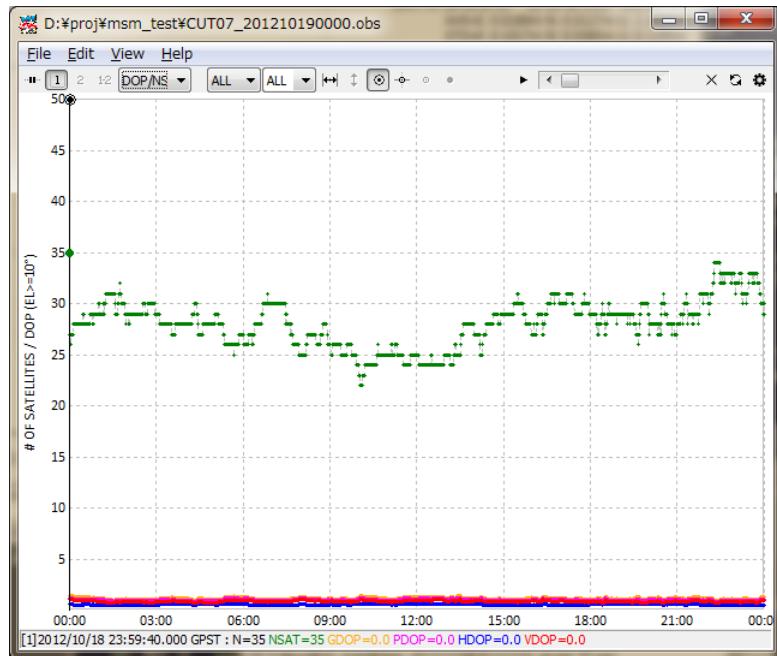


Figure 3.8-3 # of Visible Satellites and DOP Plot by RTKPLOT

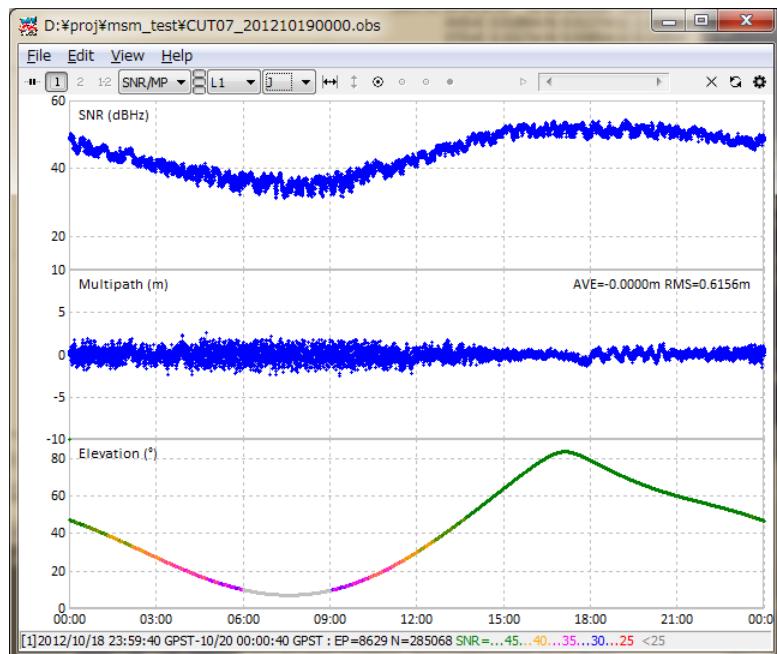


Figure 3.8-4 SNR/Multipath Plot by RTKPLOT

## RTKLIB Demo5 Manual

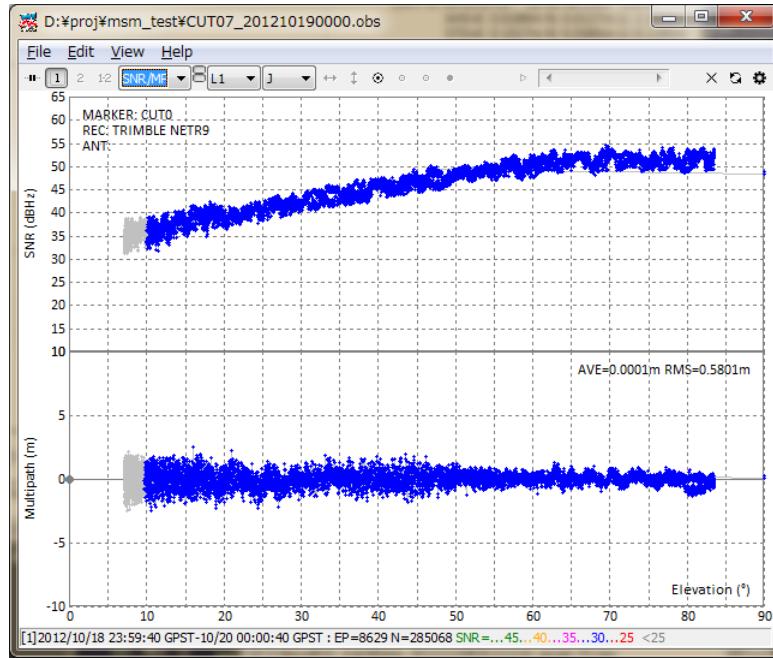


Figure 3.8-5 SNR/Multipath - Elevation Plot by RTKPLOT

- (21) You can apply the elevation mask by reading the elevation mask data by the menu "File" - "Open Elev Mask" and by set the option "Elev Mask Pattern" ON. For the format of the elevation mask data, refer the example file found in <install dir>\rtklib\_<ver>\data\elmask\_sample.txt.

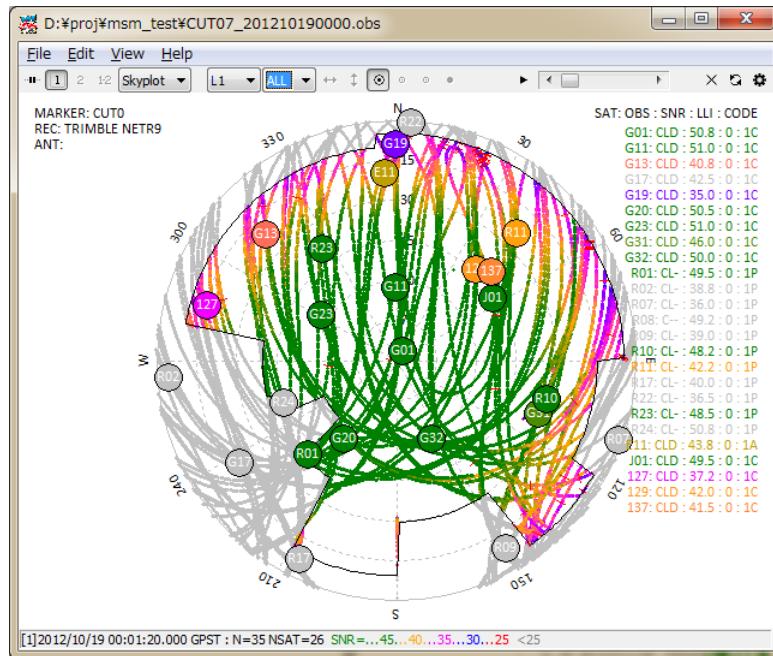


Figure 3.8-6 Elevation Mask in Skyplot by RTKPLOT

- (22) By Executing the menu "Edit" - "Obs Data Source" or "Obs Data QC", you can view the source of

solutions or the QC result as the text form.

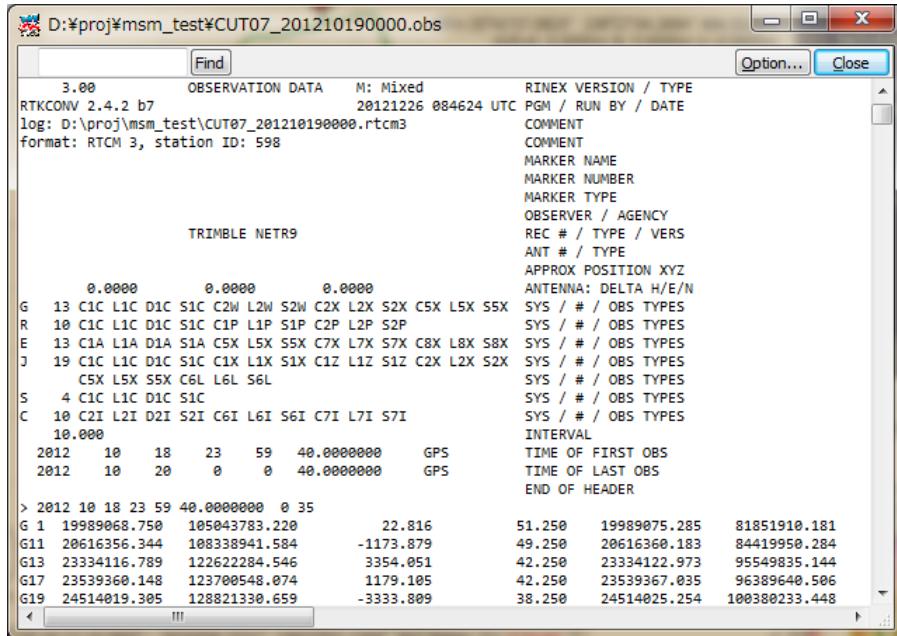
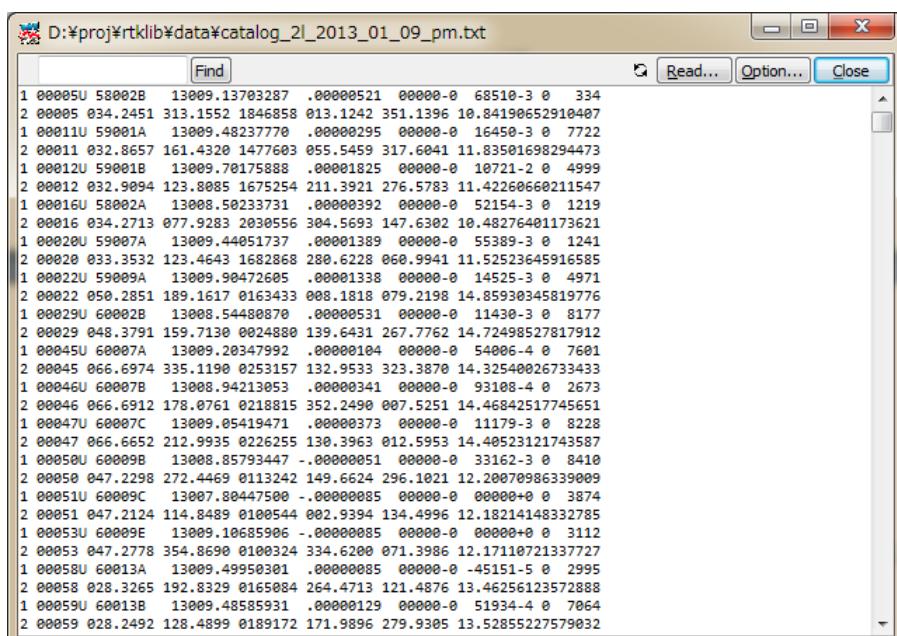


Figure 3.8-7 QC Result View by RTKPLOT

(23) In ver. 2.4.2, the satellite visibility analysis function is added. With NORAD (North American Aerospace Defense Command) TLE (two line element) data set, you can predict GNSS satellite visibility by RTKPLOT anywhere and anytime. To enable this feature, set the TLE data and the satellite number file path by "Options" dialog. The following figures show examples of these files.



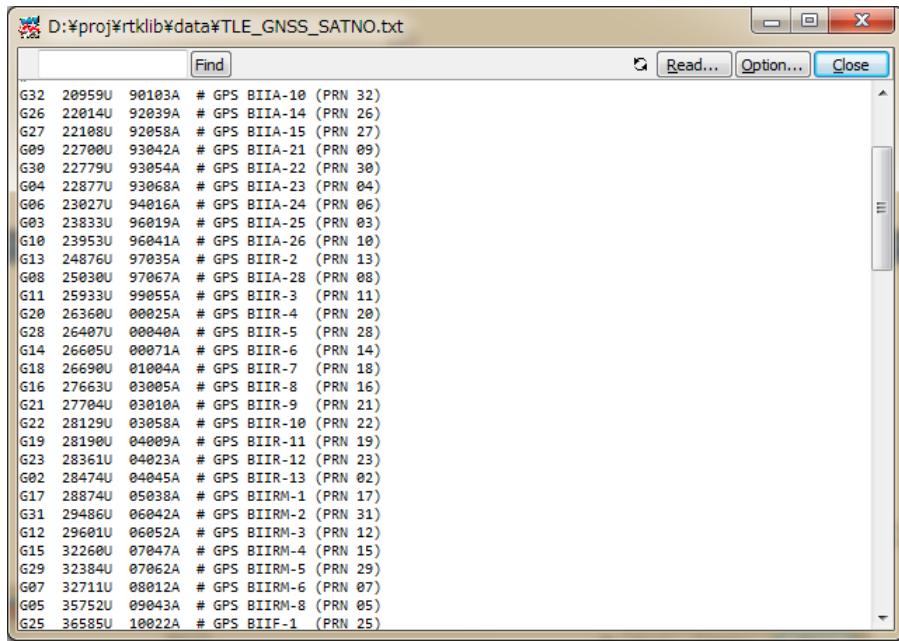


Figure 3.8-8 Examples of TLE Data (upper) and Satellite Number File (lower)

NORAD TLE data files are freely available at Celestrak (<http://celestrak.com>) or SpaceTrack (<http://www.space-track.org>). You have to download TLE data containing orbit elements of target GNSS satellites. For all of GNSS satellites, you had better to use "full catalog" for all available satellites provided by SpaceTrack. For more accurate satellite positions, you had better to use newer TLE data. Another satellite number file is used to translate the TLE satellite numbers to the GNSS satellite numbers like G23, R03, 139. A sample satellite number file can be found at `<install dir>\rtklib_<ver>\data\TLE_GNSS_SATNO.txt`. Note that the correspondence of GNSS satellite numbers with satellites is sometimes changed by the system constellation change. In this case, you shall edit and modify the satellite number file by yourself. To predict the GNSS satellite visibility, set user location as latitude, longitude and height by "Options" dialog. You have to set "Receiver Position" to "Lat/Lon/Hgt". Then execute the menu "File" - "Visibility Analysis...", you can see "Time Span/Interval" dialog.

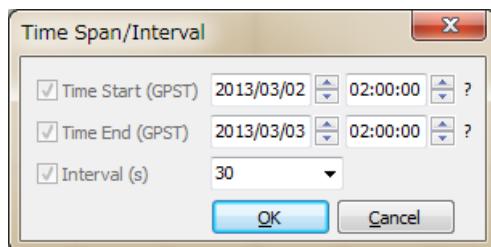


Figure 3.8-9 Time Span/Interval Dialog of RTKPLOT

By the dialog, set "Time Start", "Time End" in GPS Time, "Interval" for the analysis and push "OK" button. You can obtain a satellite visibility chart at the specified location. In the chart, the colors indicate satellite system as **GPS**, **GLONASS**, **Galileo**, **QZSS**, **BeiDou** and **SBAS** as default. You can switch the plot to Skyplot, DOP/NSat or SNR/MP/EL as same as the observation data plots.

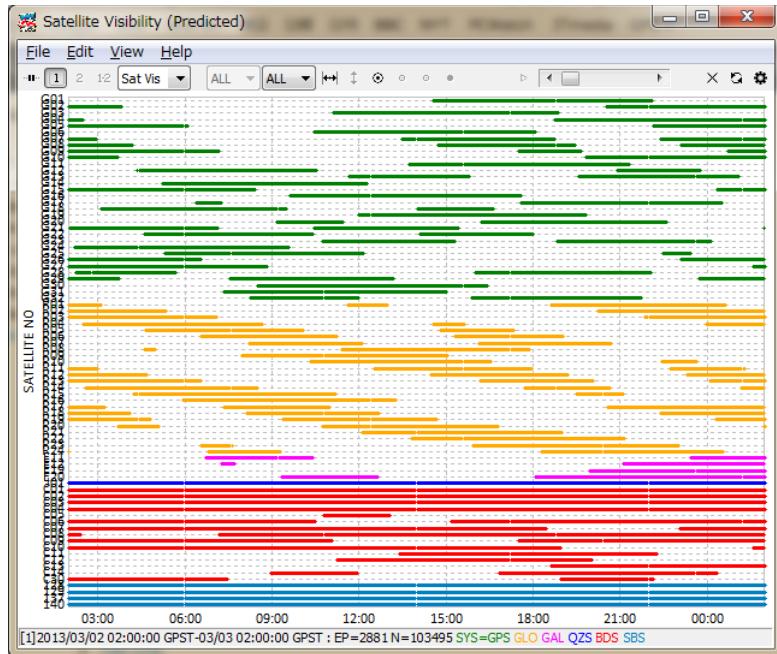


Figure 3.8-10 Satellite Visibility (Predicted) Plot by RTKPLOT

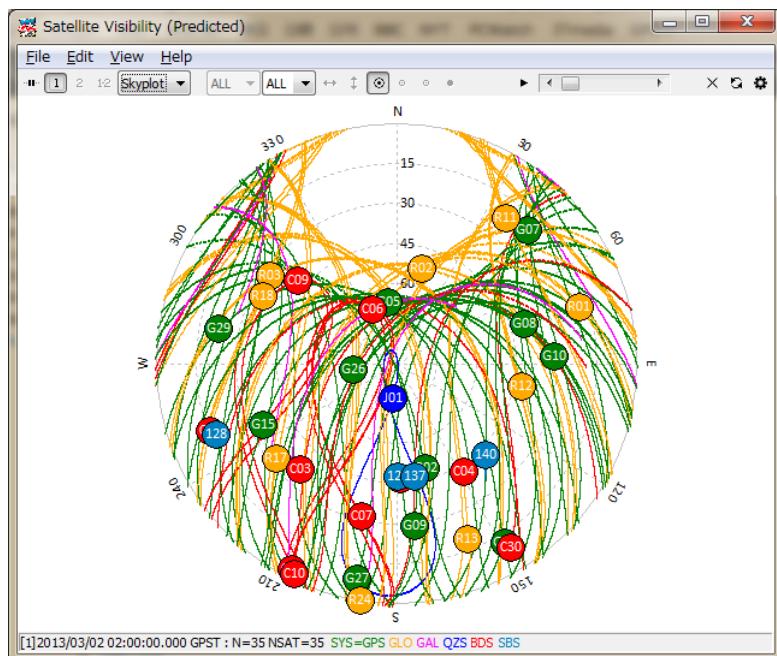


Figure 3.8-11 Skyplot (Predicted) by RTKPLOT

(24) For the plotting options for solution data by RTKPLOT and the menus provided by RTKPLOT, refer 3.7

View and Plot Solution data with RTKPLOT.

### 3.9 Download GNSS Products and Data with RTKGET

For PPP (precise point positioning) analysis, you would often need to download IGS (International GNSS service) precise GNSS products like satellite orbit and clock information. In other cases, you might want to download the observation data of a CORS (continuous operating reference stations) network from a GNSS data archive via Internet. To download these GNSS related products and data, RTKLIB offers a useful GUI download utility AP RTKGET. RTKGET is newly added in ver. 2.4.2.

- (1) Execute the binary AP file <install dir>\rtklib\_<ver>\bin\rtkget.exe. You can see the main window of RTKGET.

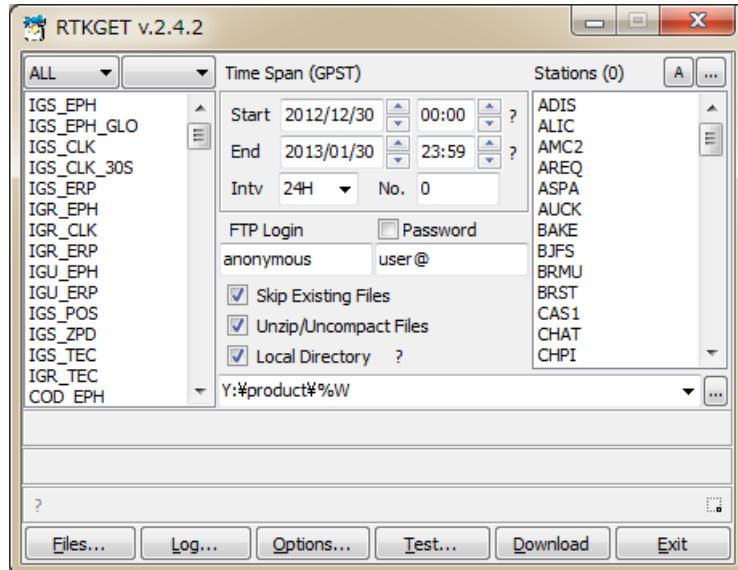


Figure 3.9-1 Main Window of RTKGET

- (2) At first, you have to configure the URL list file for GNSS data. Push **Options...** button in the main window. You can see the "Options" dialog.

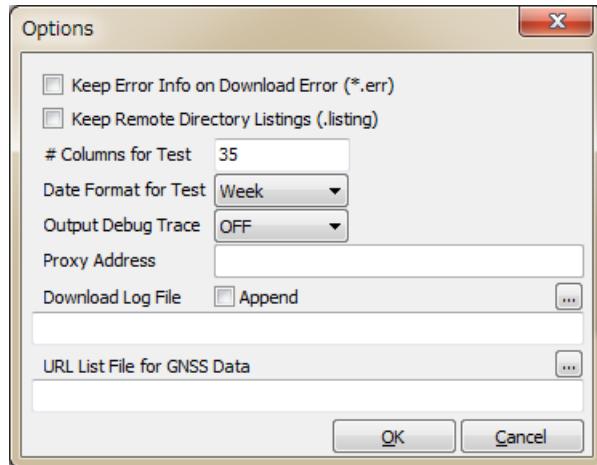


Figure 3.9-2 Options Dialog of RTKGET

- (3) Fill in the file path or select the file by pushing [...] of "URL List File for GNSS Data". Refer Appendix B.5 for the format of the file. An example of the URL list file for GNSS data is found at <install dir>\rtklib\_<ver>\data\URL\_LIST.txt. If you leave the field blank, RTKGET uses the default URL List file <install dir>\rtklib\_<ver>\data\URL\_LIST.txt.
- (4) Select download data type by the left list-box in the main window. You can select multiple data types in the list. By using combo-box above the list-box, you can filter the list by data type categories. The URL address of download data is shown at the center of the first message area. The local directory is also shown in the second message area.

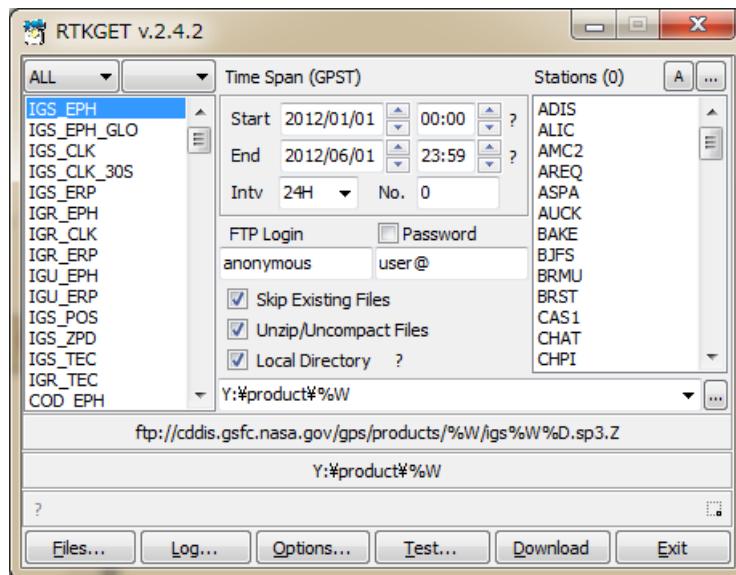


Figure 3.9-3 Selection of Download Data Type by RTKGET

- (5) Specify the time span in GPS Time with "Start", "End" and "Intv" fields. The time keywords in the URL address of download data are replaced by the time specified as the time span. If the URL address contains the keyword %N, the keyword is replaced by the number in the "No." field. To specify a sequence of numbers, use "-" to express the range like "1-99" in the "No." field.
- (6) The URL address contains keyword %S, %S or %R, the keyword is replaced by the station names selected in the right list-box "Stations". To select the station name, push [...] button at the right upper corner in the main window. You can see the "Stations" dialog. You can input or edit the station name lists in the dialog. A line of the station name lists indicate a station name to be used to replace the keywords in the URL address. To load station name list from a external text file, push [Load...] button to select the file. You can save the station name list to external text file by pushing [Save...] button. The external text file is simple text file containing station names as text lines. After input or edit the station name list, push [OK] button to close the dialog and set the station name list to the main window.

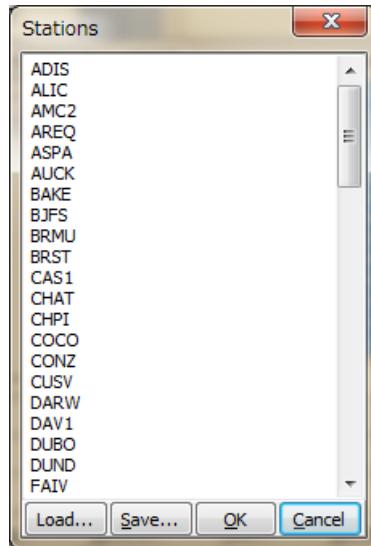


Figure 3.9-4 Stations Dialog of RTKGET

- (7) In case of FTP download, set "FTP Login" and "Password" for FTP server login. For the most of the anonymous FTP servers, you can use "anonymous" and your E-mail address for the files. Then check or uncheck the following download options:
- (a) Skip Existing Files: Set whether you skip the download or not if the local file exists
  - (b) Unzip/Uncompact Files: Set whether you unzip or uncompress the download files if these are compressed.

(c) Local Directory: Check and set the local directory path for downloaded files. Push  to select directory by the "Directory Selection" dialog . If not checked, the default download directory in the URL list file. The directory path can contain keywords to be replaced by date, time, station name and environment variable as same as URL list. Push  button to show detailed replacement in the directory path.

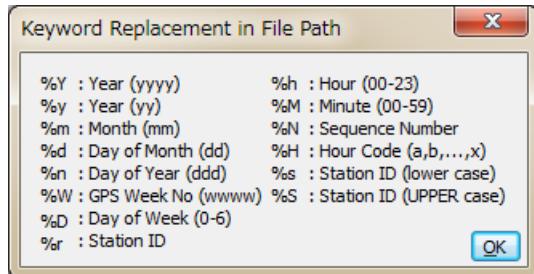


Figure 3.9-5 Keyword Replacement Dialog of RTKGET

- (8) Select station names in "Stations" list-box. Push  button to select all of stations and  button to deselect all of stations.
- (9) Push  button to start download. The third message area shows the download status indicator for each file. The indicator "\_" means in progress, "o" means download OK, "." means skipped, "x" means no data and "X" means some download error. To abort the download halfway, push  button. Even in pushing the button, the last download still in progress cannot canceled. Please wait a while to finish the last download to abort

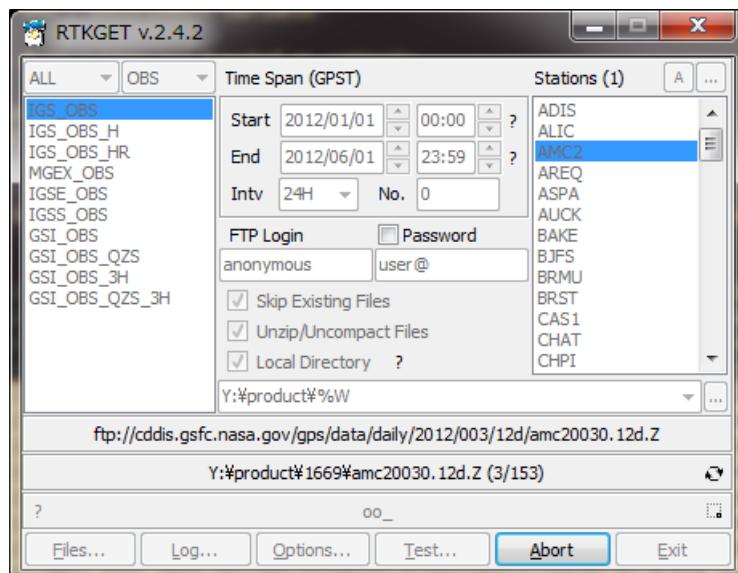
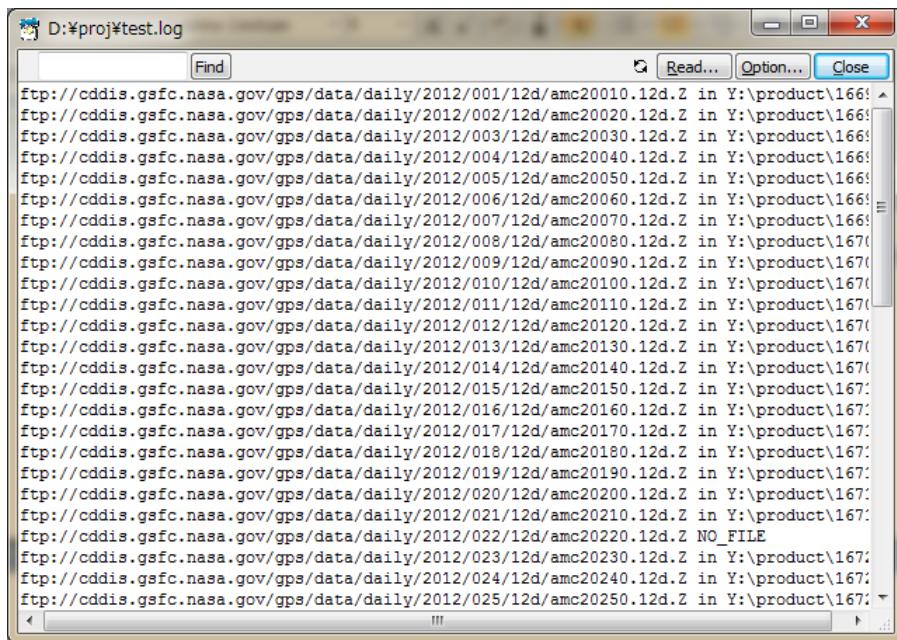


Figure 3.9-6 Data Downloading by RTKGET

- (10) After completing all of the downloads or aborting the download, push **Files...** button in the main window to execute Windows explorer showing the downloaded local directory. To see the download log, push **Log...** button. You can see the download log in "Text Viewer" window. To enable the download log function, you have to set the "Download Log File" field in the "Options" dialog. If "Append" checked for the download log, the logs are added to the existing log file if it exists. If not checked, the download log is newly generated for every tries of downloads.



- (11) To test GNSS data existence as local files before actual data download, push **Test...** button after settings as same as the actual download described above. You can see the local data availability report in the "Text Viewer" window. You can save the report by pushing **Save...** button and specify the file path in the file save dialog. The number of columns and date format of the local data availability report can be modified by the "Options" dialog.

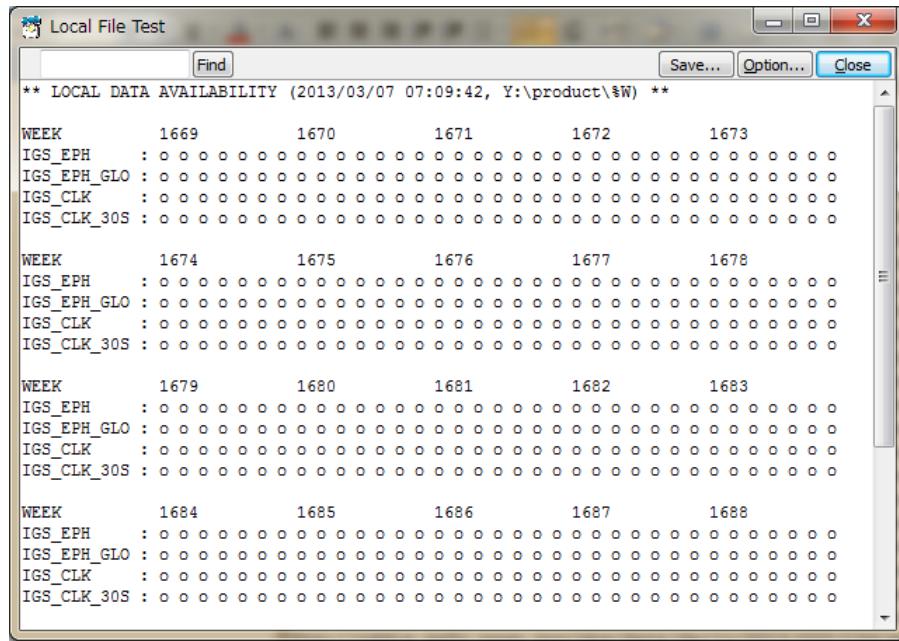


Figure 3.9-8 Local File Test View by RTKGET

- (12) By default, the detailed log for the download progress and errors is automatically deleted. To keep such error logs, check "Keep Error Info on Download Error" in the "Options" dialog. The detailed error log is saved as the file <local dir>\<file>.err for each download file. To keep remote file list file in order to analyze troubles easily, check "Keep Remote Directory Listing" as well. The remote file list is saved as the file .listing in the current directory.

### 3.10 NTRIP Browser

NTRIP (Networked Transport of RTCM via Internet Protocol) is a communication protocol to interchange GPS/GNSS related data such as receiver raw observation data, ephemerides and corrections for DGPS or RTK-GPS. NTRIP specifies the table format of so-called Source Table, which represents contents list of provided data by NTRIP servers. RTKLIB includes a simple browser for NTRIP Source Tables.

- (1) Execute the binary AP file `rtklib_<ver>\bin\srctblbrows.exe`. You can see the main window of NTRIP Source Table Browser.

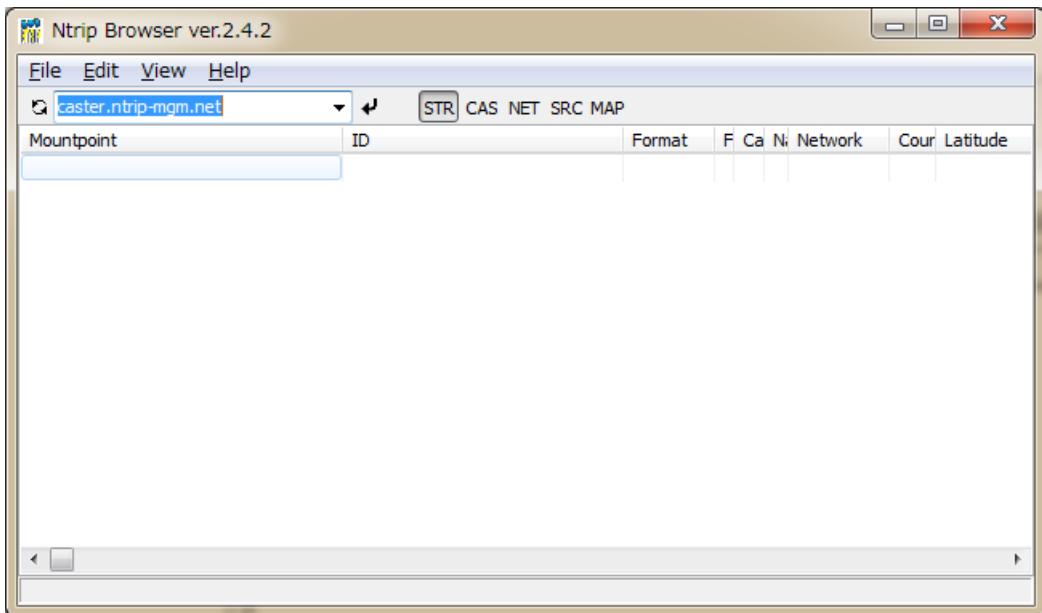


Figure 3.10-1 Main Window of NTRIP Browser

- (2) Push button upper left in the main window, leaving right pull down menu of NTRIP caster list blank. If the bottom status bar shows "connecting..." and then "update caster list", the NTRIP caster list is updated. If the pull down menu is blank, the browser acquires the NTRIP caster list from the default NTRIP info caster `rtcm-ntrip.org:2101` and update the list. To change the source of the list, fill the NTRIP caster address as the form of `<address>:<port>` in the pull down menu and push button. If you omit port number, the browser uses the default port number 2101.
- (3) Select the caster in the pull down menu and push button. If the status bar shows "source table received", the browser properly received a NTRIP Source Table from the selected NTRIP caster and shows it in the window. The status bar also indicate the error message if a problem arises.

## RTKLIB Demo5 Manual

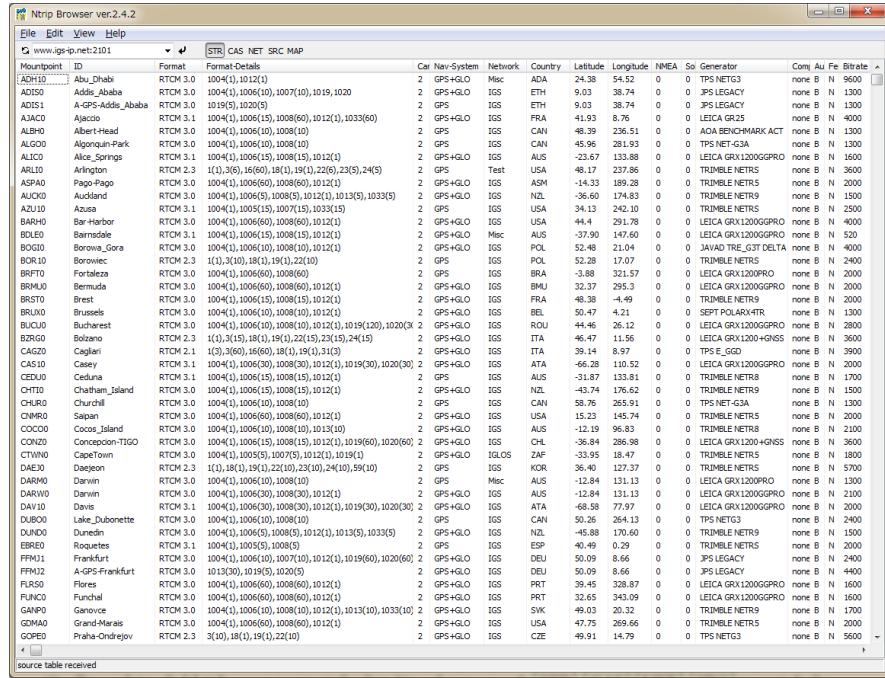


Figure 3.10-2 NTRIP Stream List View by NTRIP Browser

- (4) By pushing field title, you can sort the list by the field column. You also can push **[STR]**, **[CAS]**, **[NET]**, **[SRC]** to switch the contents of the Source Table to Stream List, Caster List, Network List and Original Source Table.

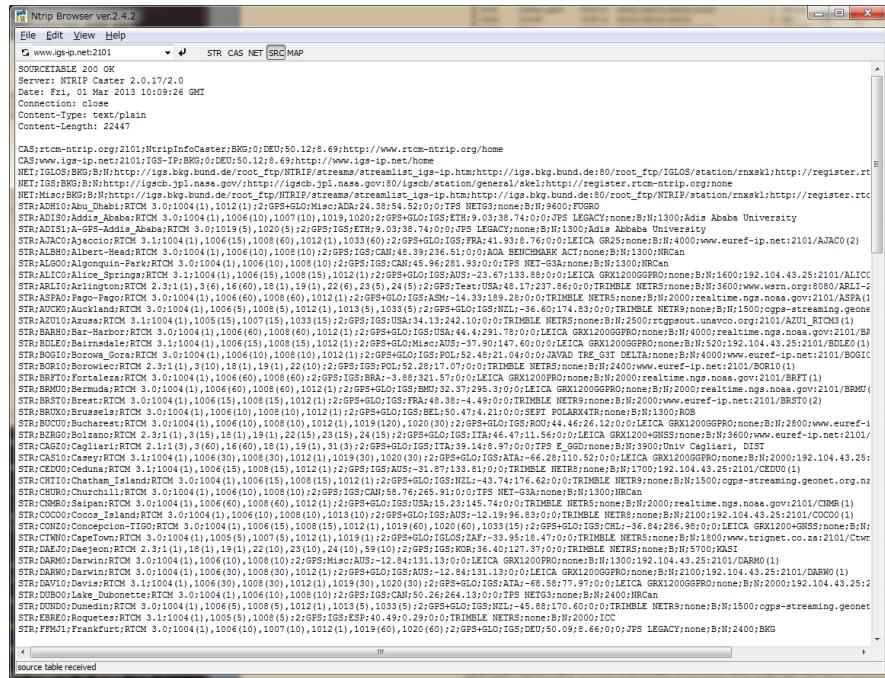


Figure 3.10-3 NTRIP Source Table View by NTRIP Browser

- (5) By pushing the [MAP] in Tool Bar, you can open the map view of the positions of the NTRIP mount points by Google Map. By click the marker in the map view, you can show the detail information of the mount point. By selecting a mount point in the NTRIP browser window, indicate the mount point position in the map by change the marker's color (red).

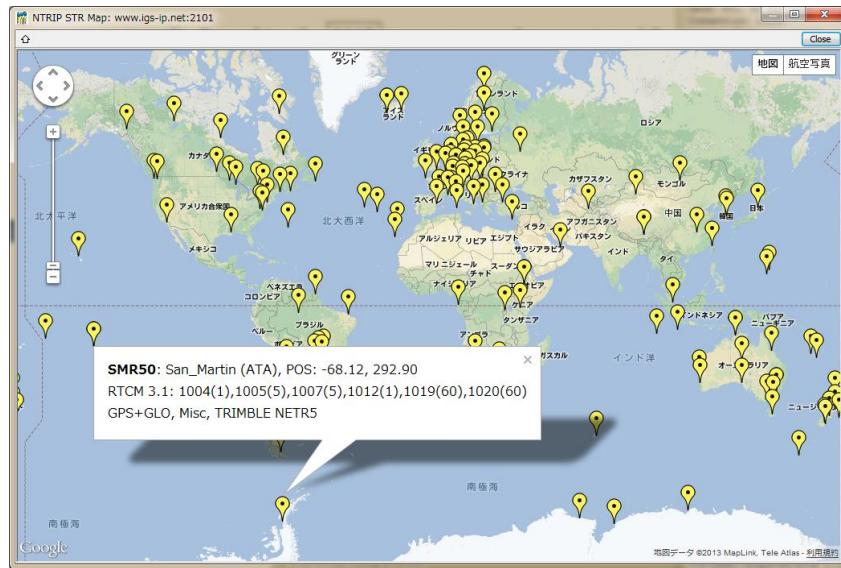


Figure 3.10-4 Map View of Station Positions by NTRIP Browser

### 3.11 Use CUI APs of RTKLIB

RTKLIB includes the following CUI APs. These CUI APs only use standard ANSI C (C89) functions and a small number of standard C libraries to ensure the portability. So you could build these CUI APs on the non-Windows environment like LINUX, UNIX, MAC OS X and so on. You can also build and execute these APs on embedded CPU like ARM. Refer Appendix A CUI Command References for these APs including command line options. For the procedure to build these APs, refer 4.2 Build CUI APs of RTKLIB. Notes that you might have to modify **makefile** to adjust the build environment but the program itself does not need to modify to port it on the most of environments.

(1) RTKRCV

Real-time Positioning. The console AP version of RTKNAVI.

(2) RNX2RTKP

Post-Processing Analysis. The console AP version of RTKPOST.

(3) POS2KML

Google Earth KML converter for solution files.

(4) CONVBIN

RINEX Converter of receiver raw data. The console AP version of RTKCONV.

(5) STR2STR

Stream Server. Console AP version of STRSVR.

## 4 Build APs or Develop User APs with RTKLIB

### 4.1 Rebuild GUI and CUI APs on Windows

To modify some functions of RTKLIB or to install your new models to RTKLIB, you may want to rebuild APs. To fix existing bugs, you also may need to rebuild APs to apply patches. To rebuild GUI and CUI APs on Windows, you need Embarcadero C++ Builder (<http://www.embarcadero.com>). RTKLIB internally only utilizes fundamental VCL (visual component library) functions provided by C++ builder. At least, the basic "starter edition" is enough to rebuild RTKLIB APs. The following instructions shows how to rebuild GUI and CUI APs on Windows.

- (1) Execute Embarcadero C++ Builder.
- (2) Execute the menu "File" - "Open Project..." of C++ builder to open the C++ builder project file of the target AP (`<app>.cbproj` or `_<app>.cbproj`, `<app>` is the target AP like `rtkpost`, `rtkplot` or `rtknavi`) in the application program directory (`<install dir>\rtklib_<ver>\app\<app>` or `<install dir>\rtklib_<ver>\app\<app>\bcc\`).
- (3) Execute the menu "Project" - "Rebuild <app>" of C++ builder to rebuild the target AP.
- (4) Execute (double click the file or input the command) the Windows batch file `install.bat` in the same directory as the project file. It copies a newly built executable binary program to the directory of RTKLIB binary programs (`<install dir>\rtklib_<ver>\bin`).
- (5) To rebuild all of the GUI APs or the CUI APs, open the C++ builder group project file `<install dir>\rtklib_<ver>\app\rtklib_winapp.gourpproj` or `<install dir>\rtklib_<ver>\app\rtklib_consapp.gourpproj`. Execute the menu "Project" - "Build All Projects" of C++ builder. Execute the batch file `install_winapp.bat` or `install_consapp.bat` in the same directory to copy them to the binary program directory.

## 4.2 Build CUI APs

To build CUI APs, you can use many C compilers like gcc. The RTKLIB package includes standard **makefile** for gcc. According to your compiler, libraries or directories, you may need to modify the **makefile** to generate APs depending to your environment.

- (1) Move to the application program directory (**rtklib\_<ver>/app/<app>**) of the target AP.

```
>> cd <install_dir>/rtklib_<ver>/app/<app>
```

- (2) Move to **gcc** directory.

```
>> cd gcc
```

- (3) Edit and modify **makefile** to adjust the file to your environment.

```
>> vi makefile
```

- (4) Execute **make** to build the AP and **make install** to install the binary file to appropriate BIN directory.

```
>> make
```

```
>> make install
```

### 4.3 Develop and Link User APs with RTKLIB

RTKLIB provide the following general purpose C-functions callable from user AP (application program). User can use these function to develop user original positioning APs.

- (1) Matrix and vector functions
- (2) Time and string functions
- (3) Coordinates transformation and geoid model
- (4) Navigation processing
- (5) Positioning models (troposphere, ionosphere, antenna PCV)
- (6) SBAS DGPS/DGNSS correction
- (7) Single point positioning
- (8) Carrier-based and code-based relative positioning
- (9) OTF integer ambiguity resolution
- (10) Receiver raw binary data input
- (11) Positioning solution/NMEA input/output
- (12) RINEX observation data/navigation message input/output
- (13) Precise ephemeris input
- (14) Stream data communication library
- (15) NTRIP (Networked Transport of RTCM via Internet Protocol) library
- (16) RTK-GPS/GNSS positioning server
- (17) RTCM 2.3 and 3.0/3.1/3.2 message handling
- (18) Downloader functions

The following instructions shows the way to utilize the library of RTKLIB in user AP.

- (1) Add the following include directive to the source program of user AP.

```
#include "rtklib.h"
```

- (2) Set the following compiler option to add RTKLIB source directory path to compiler include paths.

```
-I rtklib_<ver>\src
```

- (3) Add the necessary RTKLIB library source files to source programs set for the AP build. Refer Appendix C Library APIs for the library function list and source programs provided by RTKLIB.

## Appendix A CUI Command References

### A.1 RTKRCV

#### SYNOPSIS

```
rtkrcv [-s][-p port|-d dev] [-o file] [-t level]
```

#### DESCRIPTION

A command line version of the real-time positioning AP by RTKLIB. To start or stop RTK server, to configure options or to print solution/status, login a console and input commands. As default, stdin/stdout are used for the console. Use -p option for network login with telnet protocol. To show the available commands, type ? or help on the console. The initial processing options are loaded from default configuration file **rtkrcv.conf**. To change the file, use -o option. To configure the processing options, edit the configuration file or use set, load or save command on the console. To shutdown the program, use shutdown command on the console or send the USR2 signal to the process. For configuration file, refer B.4.

#### OPTIONS

<b>-s</b>	start RTK server on program startup
<b>-p port</b>	port number for telnet console
<b>-m port</b>	port number for monitor stream
<b>-d dev</b>	terminal device for console
<b>-o file</b>	configuration file
<b>-w pwd</b>	login password for remote console (\": no password)
<b>-r level</b>	output solution status file (0:off,1:states,2:residuals)
<b>-t level</b>	debug trace level (0:off,1-5:on)
<b>-sta sta</b>	station name for receiver dcb

#### COMMANDS

**start**

Start RTK server. No need the command if the program runs with -s option.

**stop**

Stop RTK server.

**restart**

**Restart RTK server.** If the processing options are set, execute the command to enable the changes.

**solution [cycle]**

Show solutions. Without option, only one solution is shown. With option, the solution is displayed at intervals of cycle (s). To stop cyclic display, send break (ctr-C).

**status [cycle]**

Show RTK status. Use option cycle for cyclic display.

**satellite [cycle]**

Show satellite status. Use option cycle for cyclic display.

**observ [cycle]**

Show observation data. Use option cycle for cyclic display.

**navidata [cycle]**

Show navigation data. Use option cycle for cyclic display.

**stream [cycle]**

Show stream status. Use option cycle for cyclic display.

**ssr [cycle]**

Show ssr corrections

**error**

Show error/warning messages. To stop messages, send break (ctr-C).

**option [opt]**

Show the values of processing options. Without option, all options are displayed. With option, only pattern-matched options are displayed.

**set opt [val]**

Set the value of a processing option to val. Without option val, prompt message is shown to input the value. The change of the processing option is not enabled

## RTKLIB Demo5 Manual

**before RTK server is restarted.**

**load [file]**

Load processing options from file. Without option, default file rtkrcv.conf is used. To enable the changes, restart RTK server.

**save [file]**

Save current processing options to file. Without option, default file rtkrcv.conf is used.

**log [file|off]**

Record console log to file. To stop recording the log, use option off.

**help|? [path]**

Show the command list. With option path, the stream path options are shown.

**exit**

Exit and logout console. The status of RTK server is not affected by the command.

**shutdown**

Shutdown RTK server and exit the program.

**!command [arg...]**

Execute command by the operating system shell. Do not use the interactive command.

## NOTES

Short form of a command is allowed. In case of the short form, the command is distinguished according to header characters.

## A.2 RNX2RTKP

### SYNOPSIS

```
rnx2rtkp [option ...] file file [...]
```

### DESCRIPTION

Read RINEX OBS/NAV/GNAV/HNAV/CLK, SP3, SBAS message log files and compute receiver (rover) positions and output position solutions.

The first RINEX OBS file shall contain receiver (rover) observations. For the relative mode, the second RINEX OBS file shall contain reference (base station) receiver observations. At least one RINEX NAV/GNAV/HNAV file shall be included in input files. To use SP3 precise ephemeris, specify the path in the files. The extension of the SP3 file shall be **.sp3** or **.eph**. All of the input file paths can include wild-cards (\*). To avoid command line deployment of wild-cards, use "..." for paths with wild-cards.

Command line options are as follows ([]:default). With -k option, the processing options are input from the configuration file. In this case, command line options precede options in the configuration file. For the configuration file, refer B.4.

### OPTIONS

```
-?      print help
-k file  input options from configuration file [off]
-o file  set output file [stdout]
-ts ds ts start day/time (ds=y/m/d ts=h:m:s) [obs start time]
-te de te end day/time (de=y/m/d te=h:m:s) [obs end time]
-ti tint  time interval (sec) [all]
-p mode   mode (0:single,1:dgps,2:kinematic,3:static,4:moving-base,
                  5:fixed,6:ppp-kinematic,7:ppp-static) [2]
-m mask   elevation mask angle (deg) [15]
-f freq   number of frequencies for relative mode (1:L1,2:L1+L2,3:L1+L2+L5) [2]
-v thres  validation threshold for integer ambiguity (0.0:no AR) [3.0]
-b       backward solutions [off]
-c       forward/backward combined solutions [off]
-i       instantaneous integer ambiguity resolution [off]
-h       fix and hold for integer ambiguity resolution [off]
-e       output x/y/z-ecef position [latitude/longitude/height]
-a       output e/n/u-baseline [latitude/longitude/height]
```

## RTKLIB Demo5 Manual

```
-n      output NMEA-0183 GGA sentence [off]
-g      output latitude/longitude in the form of ddd mm ss.ss [ddd.ddd]
-t      output time in the form of yyyy/mm/dd hh:mm:ss.ss [sssss.ss]
-u      output time in utc [gpst]
-d col  number of decimals in time [3]
-s sep   field separator [' ']
-r x y z  reference (base) receiver ecef pos (m) [average of single pos]
-l lat lon hgt reference (base) receiver latitude/longitude/height (deg/m)
-y level  output solution status (0:off,1:states,2:residuals) [0]
-x level  debug trace level (0:off) [0]
```

### EXAMPLES

Example 1. Kinematic Positioning, L1+L2, output Latitude/Longitude/Height to STDOUT.

```
> rnx2rtkp 07590920.05o 30400920.05o 30400920.05n
```

Example 2. Single Point Positioning, El Mask=15deg, output NMEA GGA to file out.pos

```
> rnx2rtkp -p 0 -m 15 -n -o out.pos 07590920.05o 30400920.05n
```

Example 3. Static Positioning, L1, time form yyyy/mm/dd hh:mm:ss, output X/Y/Z-ECEF positions

```
> rnx2rtkp -p 3 -f 1 -t -e 07590920.05o 30400920.05o 30400920.05n
```

Example 4. Kinematic Positioning, Instantaneous AR, validation threshold=2, comma separator

```
> rnx2rtkp -i -v 2 -s , 07590920.05o 30400920.05o 30400920.05n
```

### A.3 POS2KML

#### **SYNOPSIS**

```
pos2kml [option ...] file [...]
```

#### **DESCRIPTION**

Read position file(s) and convert it to Google Earth KML file. Each line in the input file shall contain fields of time, position fields (Latitude/Longitude/Height or X/Y/Z-ECEF), and Quality flag (option). The line started with '%', '#', ';' is treated as comment. Command options are as follows. ([]:default)

#### **OPTIONS**

```
-h      print help
-o file  output file [infile + .kml]
-c color track color
          (0:off,1:white,2:green,3:orange,4:red,5:yellow) [5]
-p color point color
          (0:off,1:white,2:green,3:orange,4:red,5:by qflag) [5]
-a      output altitude information [off]
-ag     output geodetic altitude [off]
-tg     output time stamp of gpst [off]
-tu     output time stamp of utc [gpst]
-i tint  output time interval (s) (0:all) [0]
-q qflg  output q-flags (0:all) [0]
-f n e h add north/east/height offset to position (m) [0 0 0]
-gpx    output GPX file
```

## A.4 CONVBIN

### SYNOPSIS

```
convbin [-ts y/m/d h:m:s] [-te y/m/d h:m:s] [-ti tint] [-r format] [-ro opts]
        [-f freq] [-hc comment] [-hm marker] [-hn markno] [-ht marktype]
        [-ho observ] [-hr rec] [-ha ant] [-hp pos] [-hd delta] [-v ver] [-od]
        [-os] [-x sat] [-y sys] [-d dir] [-c satid] [-o ofile] [-n nfile]
        [-g gfile] [-h hfile] [-q qfile] [-s sfile] file
```

### DESCRIPTION

Convert RTCM, receiver raw data log and RINEX file to RINEX and SBAS/LEX message file. SBAS message file complies with RTKLIB SBAS/LEX message format. It supports the following messages or files.

RTCM 2	: Type 1, 3, 9, 14, 16, 17, 18, 19, 22
RTCM 3	: Type 1002, 1004, 1005, 1006, 1010, 1012, 1019, 1020
	Type 1071-1127 (MSM except for compact msg)
NovAtel OEMV/4/6,OEMStar	: RANGECMPB, RANGEB, RAWEPHEMB, IONUTCB, RAWWASSFRAMEB
Swiftnav	:
Septentrio	:
u-blox M8T/F9P	: RXM-RAW, RXM-SFRB, RXM-RAWX
Hemisphere	: BIN76, BIN80, BIN94, BIN95, BIN96
SkyTraq S1315F	: msg0xDD, msg0xE0, msg0xDC
GW10	: msg0x08, msg0x03, msg0x27, msg0x20
Javad	: [R*], [r*], [*R], [*r], [P*], [p*], [*P], [*p], [D*], [*d],           [E*], [*E], [F*], [TC], [GE], [NE], [EN], [QE], [UO], [IO],           [WD]
NVS	: NVS NV08C BINR
BINEX	: big-endian, regular CRC, forward record (0xE2)           0x01-01,0x01-02,0x01-03,0x01-04,0x01-06,0x7f-05
RINEX	: OBS, NAV, GNAV, HNAV, LNAV, QNAV

### OPTIONS

```
file      input receiver binary log file
-ts y/m/d h:m:s  start time [all]
-te y/m/d h:m:s  end time [all]
```

```

-tr y/m/d h:m:s  approximated time for rtcm messages
-ti tint  observation data interval (s) [all]
-tt ttol  observation data epoch tolerance (s) [0.005]",
-r format log format type
    rtcm2= RTCM 2
    rtcm3= RTCM 3
    nov  = NovAtel OEMV/4/6,OEMStar
    oem3 = NovAtel OEM3
    ubx  = ublox LEA-4T/5T/6T/7T/M8/F9
    sbp  = Swift Navigation SBP
    ss2  = NovAtel Superstar II
    hemis= Hemisphere Eclipse/Crescent
    stq  = SkyTraq S1315F
    javad= Javad
    nvs  = NVS BINR
    binex= BINEX
    rinex= RINEX

-ro opt  receiver options
-f freq  number of frequencies [2]
-hc comment  rinex header: comment line
-hm marker   rinex header: marker name
-hn markno   rinex header: marker number
 ht marktype rinex header: marker type
-ho observ   rinex header: observer name and agency separated by /
-hr rec      rinex header: receiver number, type and version separated by /
-ha ant      rinex header: antenna number and type separated by /
-hp pos      rinex header: approx position x/y/z separated by /
-hd delta    rinex header: antenna delta h/e/n separated by /
-v ver       rinex version [2.11]
-od          include doppler frequency in rinex obs [off]
-os          include snr in rinex obs [off]
-oi          include iono correction in rinex nav header [off]
-ot          include time correction in rinex nav header [off]
-ol          include leap seconds in rinex nav header [off]
-halfc       half-cycle ambiguity correction [off]",
-mask [sig[,...]] signal mask(s) (sig={G|R|E|J|S|C|I}L{1C|1P|1W|...})",

```

## RTKLIB Demo5 Manual

```
-nomask [sig[,...]] signal no mask (same as above)",
-x sat    exclude satellite
-y sys    exclude systems (G:GPS,R:GLO,E:Galileo,J:QZSS,S:SBAS,C:BeiDou)
-d dir    output directory [same as input file]
-c staid  use RINEX file name convention with staid [off]
-o ofile  output RINEX OBS file
-n nfile  output RINEX NAV file
-g gfile  output RINEX GNAV file
-h hfile  output RINEX HNAV file
-q qfile  output RINEX QNAV file
-l lfile  output RINEX LNAV file
-b cfile  output RINEX CNAV file",
-i ifile  output RINEX INAV file",
-s sfile  output SBAS message file
-trace level output trace level [off]",
```

If any output file specified, default output files (<file>.obs, <file>.nav, <file>.gnav, <file>.hnav, <file>.qnav, <file>.lnav and <file>.sbs) are used.

If receiver type is not specified, type is recognized by the input file extension as follows.

```
*.rtcm2    RTCM 2
*.rtcm3    RTCM 3
*.gps      NovAtel OEMV/4/6,OEMStar
*.ubx      u-blox LEA-4T/5T/6T/8T/8/9
*.bin      Hemisphere Eclipse/Crescent
*.stq      SkyTraq S1315F
*.jps      Javad
*.bnx,*binex BINEX
*.obs,*.*o RINEX OBS
```

## A.5 STR2STR

### SYNOPSIS

```
str2str -in stream[#...] -out stream[#...] [-out stream[#...]...] [options]
```

### DESCRIPTION

Input data from a stream and divide and output them to multiple streams. The input stream can be serial, TCP client, TCP server, NTRIP client, or file. The output stream can be serial, TCP client, TCP server, NTRIP server, or file. **str2str** is a resident type application. To stop it, type Ctrl-C in console if run foreground or send signal SIGINT for background process. if run foreground or send signal SIGINT for background process. if both of the input stream and the output stream follow **#format**, the format of input messages are converted to output. To specify the output messages, use **-msg** option.

Command options are as follows.

### OPTIONS

```
-in stream[#format] input stream path and format
-out stream[#format] output stream path and format

stream path
    serial : serial://port[:brate[:bsize[:parity[:stopb[:fctr]]]]][#tcp_port]
    tcp server : tcpsvr://:port
    tcp client : tcpcli://addr[:port]
    ntrip client : ntrip://[user[:passwd>@]addr[:port][/:mntpnt]
    ntrip server : ntrips://[:passwd@]addr[:port][/:mntpnt[:str]] (only out)
    ntrip caster : ntricp://[user:passwd@][:port]/mntpnt[:srctbl] (only out)
    file : [file://]path[::T][::+start][::xseppd][::S=swap]

format
    rtcm2 : RTCM 2 (only in)
    rtcm3 : RTCM 3 (in and out)
    nov : NovAtel OEMV/4/6,OEMStar (only in)
    swiftnav : Swiftnav (only in)
    ubx : ublox LEA-4T/5T/6T/8/9 (only in)
    hemis : Hemisphere Eclipse/Crescent (only in)
    stq : SkyTraq S1315F (only in)
```

RTKLIB Demo5 Manual

```
javad      : Javad (only in)
sbf        : Septentrio (only in)
nvs        : NVS BINR (only in)
binex      : BINEX (only in)

-msg type[(tint)][,type[(tint)]...]
          rtcm message types and output intervals (s)

-sta sta    station id
-opt opt     receiver dependent options
-s msec      timeout time (ms) [10000]
-r msec      reconnect interval (ms) [10000]
-n msec      nmea request cycle (m) [0]
-f sec       file swap margin (s) [30]
-c file      input commands file [no]
-c1 file     output 1 commands file [no]
-c2 file     output 2 commands file [no]
-c3 file     output 3 commands file [no]
-c4 file     output 4 commands file [no]
-p lat lon hgt station position (latitude/longitude/height) (deg,m)
-px x y z    station position (x/y/z-ecef) (m)
-a antinfo   antenna info (separated by ,)
-i rcvinfo   receiver info (separated by ,)
-o e n u     antenna offset (e,n,u) (m)
-l local_dir ftp/http local directory []
-x proxy_addr http/ntrip proxy address [no]
-b str_no    relay back messages from output str to input str [no]
-t level     trace level [0]
-fl file     log file [str2str.trace]
-h          print help
```

## Appendix B File Formats

### B.1 Positioning Solution File

#### DESCRIPTION

A positioning solution file is just a text file as output of RTKNAVI or RTKPOST. The file is separated to records or lines by CR/LF. Each records consists of fields. The following table shows the format of the positioning solution file.

No	Record/Field	Description	Notes
1	File header	<p>The lines starting with "%" are header lines. The header lines contains some additional information or processing options as follows.</p> <ul style="list-style-type: none"> <li>% <b>program</b>: program version</li> <li>% <b>inp file</b>: Input file path</li> <li>% <b>obs start</b>: Observation data start time in GPS time</li> <li>% <b>obs end</b>: Observation data end time in GPS time</li> <li>% <b>pos mode</b>: Positioning mode option</li> <li>% <b>freqs</b>: Frequencies option</li> <li>% <b>solution</b>: Solution type option</li> <li>% <b>elev mask</b>: Elevation mask angle option</li> <li>% <b>snr mask</b>: SNR mask option</li> <li>% <b>ionos est</b>: Ionospheric parameter estimation option</li> <li>% <b>tropos est</b>: Tropospheric parameters estimation option</li> <li>% <b>amb res</b>: Integer ambiguity resolution options</li> <li>% <b>val thres</b>: Integer ambiguity validation option</li> <li>% <b>ref pos</b>: position of the antenna of the base station</li> </ul>	
2	Field indicator	Field indicator starting with "%" line follows after File header. To recognize the field formats, RTKLIB uses these lines. Do not delete them.	
3	Solution body	Solution body consists of the following fields. The field contents are varied according to the positioning options.	
(1)	Time	<p>The epoch time of the solution indicating the true receiver signal reception time (not indicates the time by receiver clock). The format is varied to the options.</p> <p><b>yyyy/mm/dd HH:MM:SS.SSS :</b> Calendar time in GPST, UTC or JST, the time system is indicated in Field indicator</p> <p><b>WWWW SSSSSSS.SSS :</b> GPS week and TOW (time of week) in seconds.</p>	
(2)	Receiver Position	The rover receive antenna or marker position estimated varied according to the positioning options.	

No	Record/Field	Description	Notes
		<b>+ddd.ddddddd dddd.ddddddd hhh.hhh :</b> Latitude, longitude in degrees and height in m. Minus value means south latitude or west longitude. The height indicates ellipsoidal or geodetic according to the positioning options.	
		<b>+ddd mm ss.sss +ddd mm ss.sss hhh.hhh :</b> Latitude, longitude in degree, minute and second and height in m.	
		<b>+xxxxxxxxx.xxxx +yyyyyyyyy.yyyy +zzzzzzz.zzzz :</b> X/Y/Z components of ECEF frame in m.	
		<b>+eeeeeee.eeee +nnnnnnnnn.nnnn +uuuuuuuuu.uuuu :</b> E/N/U components of baseline vector in m. The local coordinate is referenced to the rover position.	
(3)	Quality flag (Q)	The flag which indicates the solution quality. <b>1</b> : Fixed, solution by carrier-based relative positioning and the integer ambiguity is properly resolved. <b>2</b> : Float, solution by carrier-based relative positioning but the integer ambiguity is not resolved. <b>3</b> : Reserved <b>4</b> : DGPS, solution by code-based DGPS solutions or single point positioning with SBAS corrections <b>5</b> : Single, solution by single point positioning	
(4)	Number of valid satellites (ns)	The number of valid satellites for solution estimation.	
(5)	Standard deviations (sdn, sde, sdu, sdne, sdeu, sdun)	The estimated standard deviations of the solution assuming a priori error model and error parameters by the positioning options. The sdn, sde or sdu means N (north), E (east) or U (up) component of the standard deviations in m. The absolute value of sdne, sdeu or sdun means square root of the absolute value of NE, EU or UN component of the estimated covariance matrix. The sign represents the sign of the covariance. With all of the values, user can reconstruct the full covariance matrix.	
(6)	Age of differential (age)	The time difference between the observation data epochs of the rover receiver and the base station in second.	
(7)	Ratio factor (ratio)	The ratio factor of "ratio-test" for standard integer ambiguity validation strategy. The value means the ratio of the squared sum of the residuals with the second best integer vector to with the best integer vector.	
(8)	Velocity (optional) (vn, ve, vu, sdvn, sdve sdvu, sdvne sdveu, sdvun )	Velocity components and standard deviations (Enable from options menu)	

**EXAMPLE**

```
% program : RTKLIB ver.2.3.0b
% inp file : G:\rtklibtest\20090831\omre196a.09o
% inp file : G:\rtklibtest\20090831\tevc196a.09o
% inp file : G:\rtklibtest\20090831\omre196a.09n
% obs start : 2009/07/15 07:10:00.0 GPST (week1540 285000.0s)
% obs end : 2009/07/15 07:59:50.0 GPST (week1540 287990.0s)
% pos mode : kinematic
% freqs : L1+L2
% solution : forward
% elev mask : 15.0 deg
% snr mask : 0.0 dBHz
% ionos est : on
% tropo est : on
% amb res : continuous
% val thres : 3.0
% ref pos : 32.574831620 -117.126551777 -28.1471
%
% (lat/lon/height=WGS84/ellipsoidal,Q=1:fix,2:float,4:dgps,5:single,ns=# of satellites)
% GPST           latitude(deg) longitude(deg) height(m) Q ns sdn(m) sde(m)
sdu(m)  sdne(m)  sdeu(m)  sdun(m) age(s) ratio
2009/07/15 07:10:00.000 32.560273272 -116.953525346 118.6783 1 10 0.0186 0.0202
0.0899 -0.0072 0.0089 -0.0249 0.00 4.5
2009/07/15 07:10:10.000 32.560273266 -116.953525340 118.6877 1 10 0.0144 0.0154
0.0776 -0.0058 0.0082 -0.0199 0.00 5.3
2009/07/15 07:10:20.000 32.560273262 -116.953525365 118.6853 1 10 0.0124 0.0131
0.0720 -0.0051 0.0078 -0.0173 0.00 5.3
2009/07/15 07:10:30.000 32.560273251 -116.953525345 118.6825 1 10 0.0111 0.0117
0.0686 -0.0046 0.0075 -0.0157 0.00 5.6
2009/07/15 07:10:40.000 32.560273275 -116.953525412 118.6827 1 10 0.0103 0.0108
0.0662 -0.0043 0.0073 -0.0146 0.00 4.7
2009/07/15 07:10:50.000 32.560273277 -116.953525429 118.6812 1 10 0.0097 0.0102
0.0644 -0.0041 0.0071 -0.0138 0.00 4.1
2009/07/15 07:11:00.000 32.560273249 -116.953525449 118.6817 1 10 0.0092 0.0097
0.0630 -0.0039 0.0069 -0.0132 0.00 4.2
2009/07/15 07:11:10.000 32.560273271 -116.953525464 118.6729 1 10 0.0088 0.0093
0.0618 -0.0038 0.0067 -0.0127 0.00 5.2
2009/07/15 07:11:20.000 32.560273246 -116.953525468 118.6772 1 10 0.0085 0.0089
0.0607 -0.0037 0.0066 -0.0123 0.00 6.1
2009/07/15 07:11:30.000 32.560273219 -116.953525461 118.6733 1 10 0.0083 0.0087
0.0598 -0.0036 0.0065 -0.0119 0.00 7.9
2009/07/15 07:11:40.000 32.560273216 -116.953525478 118.6771 1 10 0.0081 0.0085
0.0590 -0.0035 0.0064 -0.0117 0.00 9.0
2009/07/15 07:11:50.000 32.560273206 -116.953525489 118.6726 1 10 0.0079 0.0083
0.0582 -0.0035 0.0062 -0.0114 0.00 8.6
2009/07/15 07:12:00.000 32.560273201 -116.953525497 118.6744 1 10 0.0078 0.0081
0.0575 -0.0034 0.0061 -0.0112 0.00 7.5
2009/07/15 07:12:10.000 32.560273212 -116.953525455 118.6731 1 10 0.0077 0.0080
0.0568 -0.0034 0.0060 -0.0110 0.00 7.9
```

## B.2 SBAS Log File

## **DESCRIPTION**

A SBAS log file is output of RTKCONV, that is a text file in which a line contains a SBAS message captured by the GPS/GNSS receiver. The following table shows the format of the SBAS log file.

No	Record/Field	Description	Notes
1	SBAS messages	A line contains a SBAS navigation data frame, which consists of the following fields.	
(1)	GPS week number	GPS week number of SBAS navigation data frame.	
(2)	Time of week	Time of week of SBAS navigation data frame in seconds.	
(3)	PRN number	PRN number of SBAS satellite transmitting the navigation data	
(4)	Message type	The type ID of the SBAS message in the frame (0 - 63). Refer SBAS specifications for details: RTCA/DO-229C, Minimum operational performance standards for Global Positioning System/Wide Area Augmentation system airborne equipment	
(5)	Separator	:	
(6)	SBAS message	Hexadecimal dump of a 226-bit SBAS message without 24-bit parity field. The message tail is 0-padded to align to 8-bit boundary. Refer SBAS specifications for the detailed message format.	

## **EXAMPLE**

```
1471 349019 137 26 : C66A0431E17E0A704741F80DC05E827815C0EF09F877C2FC15E096E000
1471 349020 137 4 : 53119FFC001FFDFDFFDFFF9C001FFDFDFDFFDFFF9BBBBB33FFFFC0
1471 349020 129 4 : 53129FFFFFF5FFDFDFDFFF3FA9FFDFDFDFFDFFF9BBBBB33FFFFC0
```

## B.3 Solution Status File

### DESCRIPTION

A solution status file is output of RTKNAVI or RTKPOST, that is a text file which contains the internal status of the positioning process. The internal status include estimated states of Kalman filter and residuals of measurements to analyze the solution quality. The following table shows the format of the solution status file.

No	Record/Field	Description	Notes
1	Position States	<p>Estimated rover position in the filter. The format of a record is as follows.</p> <pre>\$POS,week,tow,stat,posx,posy,posz,posxf,posyf,poszf week/tow : gps week no/time of week (s) stat   : solution status posx/posy/posz  : position x/y/z ecef (m) float posxf/posyf/poszf : position x/y/z ecef (m) fixed</pre>	
2	Velocity/ Acceleration States	<p>Estimated rover velocity and acceleration in the filter. The format of a record is as follows.</p> <pre>\$VELACC,week,tow,stat,vele,veln,velu,acce,accn,accu,velef,velnf, \ veluf,accef,accnf,accuf week/tow : gps week no/time of week (s) stat   : solution status vele/veln/velu  : velocity e/n/u (m/s) float acce/accn/accu  : acceleration e/n/u (m/s^2) float velef/velnf/veluf : velocity e/n/u (m/s) fixed accef/accnf/accuf : acceleration e/n/u (m/s^2) fixed</pre>	
3	Receiver Clock-bias States	<p>Estimated receiver clock bias parameters. The format of a record is as follows.</p> <pre>\$CLK,week,tow,stat,rcv,clk1,clk2,clk3,clk4,cmn_bias week/tow : gps week no/time of week (s) stat   : solution status clk1   : receiver clock bias GPS (ns) clk2   : receiver clock bias GLO-GPS (ns) clk3   : receiver clock bias GAL-GPS (ns) clk4   : receiver clock bias BDS-GPS (ns) cmn_bias : common phase bias removed from all</pre>	
4	Ionosphere Parameter States	<p>Estimated ionosphere parameter (vertical L1 ionosphere delay difference). The format of a record is as follows.</p> <pre>\$ION,week,tow,stat,sat,az,el,ion,ion-fixed week/tow : gps week no/time of week (s) stat   : solution status</pre>	

No	Record/Field	Description	Notes
		sat : satellite id az/el : azimuth/elevation angle(deg) ion : vertical ionospheric delay L1 (m) float ion-fixed: vertical ionospheric delay L1 (m) fixed	
5	Troposphere Parameter States	Estimated troposphere parameter (vertical troposphere delay residual). The format of a record is as follows.  \$TROP,week,tow,stat,rcv,ztd,ztdf week/tow : gps week no/time of week (s) stat : solution status rcv : receiver (1:rover,2:base station) ztd : zenith total delay (m) float ztdf : zenith total delay (m) fixed	
6	Receiver H/W bias States	Estimated GLONASS receiver H/W bias difference. The format of a record is as follows.  \$HWBIAS,week,tow,stat,frq,bias,biasf week/tow : gps week no/time of week (s) stat : solution status frq : frequency (1:L1,2:L2,3:L5,...) bias : h/w bias coefficient (m/MHz) float biasf : h/w bias coefficient (m/MHz) fixed	
7	Residuals	Residuals of pseudorange and carrier-phase observables. The format of a record is as follows.  \$SAT,week,tow,sat,frq,az,el,resp,resc,vsat,snr,fix,slip,lock,outc,\ slipc,rejc,icbias,bias,bias_var,lambda week/tow : gps week no/time of week (s) sat/frq : satellite id/frequency (1:L1,2:L2,3:L5,...) az/el : azimuth/elevation angle (deg) resp : pseudorange residual (m) resc : carrier-phase residual (m) vsat : valid data flag (0:invalid,1:valid) snr : signal strength (dBHz) fix : ambiguity flag (0:no data,1:not part of AR set,2:part of AR set,3:part of hold set) slip : cycle-slip flag (bit1:slip,bit2:parity unknown) lock : carrier-lock count outc : data outage count slipc : cycle-slip count rejc : data reject (outlier) count icbias : interchannel bias (GLONASS) bias : phase bias bias_var : variance of phase bias lambda : wavelength	

**EXAMPLE**

```
$POS,1557,432000.000,2,-3869295.9628,3436570.2567,3717367.6546,0.0000,0.0000,0.0000
$HWBIAS,1557,432000.000,2,1,-0.3503,0.0000
$HWBIAS,1557,432000.000,2,2,0.0108,0.0000
$SAT,1557,432000.000,3,1,253.2,64.3,0.3219,-0.0006,1,48,1,1,1,0,1,0
$SAT,1557,432000.000,3,2,253.2,64.3,-0.0629,-0.0006,1,33,1,1,1,0,1,0
```

## RTKLIB Demo5 Manual

```
$SAT,1557,432000.000,13,1,298.4,24.1,-0.6732,0.0003,1,42,1,1,1,0,1,0  
$SAT,1557,432000.000,13,2,298.4,24.1,0.8081,0.0003,1,17,1,1,1,0,1,0  
$SAT,1557,432000.000,16,1,42.0,59.5,0.5037,-0.0005,1,47,1,1,1,0,1,0  
$SAT,1557,432000.000,16,2,42.0,59.5,-0.5170,-0.0005,1,30,1,1,1,0,1,0  
$SAT,1557,432000.000,19,1,229.8,39.0,-0.1948,-0.0003,1,44,1,0,1,0,0,0  
$SAT,1557,432000.000,19,2,229.8,39.0,-0.0806,-0.0003,1,28,1,1,1,0,1,0  
$SAT,1557,432000.000,21,1,61.1,28.1,-1.0704,0.0001,1,41,1,1,1,0,1,0  
$SAT,1557,432000.000,21,2,61.1,28.1,1.0139,0.0001,1,19,1,1,1,0,1,0  
$SAT,1557,432000.000,23,1,257.9,29.9,-1.3258,-0.0000,1,45,1,1,1,0,1,0  
$SAT,1557,432000.000,23,2,257.9,29.9,0.4155,0.0000,1,23,1,1,1,0,1,0  
$SAT,1557,432000.000,25,1,317.0,24.7,0.8868,0.0002,1,41,1,1,1,0,1,0  
$SAT,1557,432000.000,25,2,317.0,24.7,0.1811,0.0003,1,19,1,1,1,0,1,0  
$SAT,1557,432000.000,31,1,145.1,32.5,0.6140,-0.0001,1,44,1,1,1,0,1,0  
$SAT,1557,432000.000,31,2,145.1,32.5,-0.2397,-0.0001,1,26,1,1,1,0,1,0  
$SAT,1557,432000.000,R9,1,105.7,78.1,-0.1172,-0.0001,1,45,1,1,1,0,1,0  
$SAT,1557,432000.000,R9,2,105.7,78.1,0.0000,0.0000,0,0,0,0,0,1,0,0  
$SAT,1557,432000.000,R10,1,331.5,41.7,-0.1425,0.0002,1,43,1,1,1,0,1,0  
$SAT,1557,432000.000,R10,2,331.5,41.7,0.0349,0.0001,1,30,1,1,1,0,1,0  
$SAT,1557,432000.000,R19,1,18.6,61.2,-0.7708,-0.0000,1,46,1,1,1,0,1,0  
$SAT,1557,432000.000,R19,2,18.6,61.2,0.1898,-0.0001,1,39,1,0,1,0,0,0  
$SAT,1557,432000.000,R20,1,235.7,55.6,1.0305,-0.0000,1,42,1,1,1,0,1,0  
$SAT,1557,432000.000,R20,2,235.7,55.6,-0.2247,-0.0001,1,39,1,1,1,0,1,0
```

## B.4 Configuration File

### DESCRIPTION

A configuration file containing processing options, solution options and file options for RTKNAVI, RTKPOST, RTKRCV and RNX2RTKP. That is a text file which contains the **Keyword = Value** form records indicating the various options. For enumeration values, the selectable value is either of a number (0,1,2,...) or an enumeration label (off, on, ...). The line starting with # and the texts after # in a line are treated as comments. For the contents of the configuration file, refer 3.5 Configure Positioning Options for RTKNAVI and RTKPOST.

### EXAMPLE

```
# RTKNAVI options (2013/03/01 10:41:04, v.2.4.2)

pos1-posmode      =single      #
(0:single,1:dgps,2:kinematic,3:static,4:movingbase,5:fixed,6:ppp-kine,7:ppp-static)
pos1-frequency    =11+12     # (1:11,2:11+12,3:11+12+15)
pos1-solttype     =forward    # (0:forward,1:backward,2:combined)
pos1-elmask       =10        # (deg)
pos1-snrmask_r    =off        # (0:off,1:on)
pos1-snrmask_b    =off        # (0:off,1:on)
pos1-snrmask_L1   =0,0,0,0,0,0,0,0
pos1-snrmask_L2   =0,0,0,0,0,0,0,0
pos1-snrmask_L5   =0,0,0,0,0,0,0,0
pos1-dynamics     =off        # (0:off,1:on)
pos1-tidecorr     =off        # (0:off,1:on)
pos1-ionoopt      =brdc       # (0:off,1:brdc,2:sbas,3:dual-freq,4:est-stec,5:ionex-
tec,6:qzs-brdc,7:qzs-lex,8:vtec_sf,9:vtec_ef,10:gtec)
pos1-tropopt      =saas       # (0:off,1:saas,2:sbas,3:est-ztd,4:est-ztdgrad)
pos1-sateph      =brdc       #
(0:brdc,1:precise,2:brdc+sbas,3:brdc+ssrapc,4:brdc+ssrcom)
pos1-posopt1      =on         # (0:off,1:on)
pos1-posopt2      =on         # (0:off,1:on)
pos1-posopt3      =on         # (0:off,1:on)
pos1-posopt4      =on         # (0:off,1:on)
pos1-posopt5      =off        # (0:off,1:on)
pos1-exclsats    =
=                   # (prn ...)
pos1-navsys       =63         # (1:gps+2:sbas+4:glo+8:gal+16:qzs+32:comp)
pos2-armode       =fix-and-hold # (0:off,1:continuous,2:instantaneous,3:fix-and-hold)
pos2-gloarmode   =off        # (0:off,1:on,2:autocal)
pos2-arthres     =3
pos2-arlockcnt   =0
pos2-arelmask    =20        # (deg)
pos2-arminfix    =0
pos2-elmaskhold  =0        # (deg)
pos2-aroutcnt   =5
pos2-maxage      =30        # (s)
pos2-sliptres   =0.05      # (m)
pos2-rejionno   =30        # (m)
pos2-rejgdop    =30
pos2-niter       =1
pos2-baselens   =0        # (m)
pos2-basesig    =0        # (m)
out-solformat   =llh       # (0:llh,1:xyz,2:enu,3:nmea)
out-outhead     =off        # (0:off,1:on)
out-outopt      =off        # (0:off,1:on)
out-timesys    =gpst       # (0:gpst,1:utc,2:jst)
out-timeform   =hms        # (0:tow,1:hms)
```

## RTKLIB Demo5 Manual

```

out-timendec      =3
out-degform       =deg      # (0:deg,1:dms)
out-fieldsep      =
out-height        =geodetic # (0:ellipsoidal,1:geodetic)
out-geoid          =internal # (0:internal,1:egm96,2:egm08_2.5,3:egm08_1,4:gsi2000)
out-solstatic     =all      # (0:all,1:single)
out-nmeaintv1     =1        # (s)
out-nmeaintv2     =1        # (s)
out-outstat       =off      # (0:off,1:state,2:residual)
stats-eratio1     =300
stats-eratio2     =300
stats-errphase    =0.003   # (m)
stats-errphaseel  =0.003   # (m)
stats-errphasebl  =0        # (m/10km)
stats-errdoppler  =1        # (Hz)
stats-stdbias     =30      # (m)
stats-stdiono     =0.03    # (m)
stats-stdtrop     =0.3     # (m)
stats-prnaccelh  =10      # (m/s^2)
stats-prnaccelv  =10      # (m/s^2)
stats-prnbias    =0.0001  # (m)
stats-prniono    =0.001   # (m)
stats-prntrop    =0.0001  # (m)
stats-clkstab    =5e-12   # (s/s)
ant1-postype     =llh     # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant1-pos1         =90      # (deg|m)
ant1-pos2         =0        # (deg|m)
ant1-pos3         =-6335367.6285 # (m|m)
ant1-anttype     =NOV702GG
ant1-antdele     =0        # (m)
ant1-antdeln     =0        # (m)
ant1-antdelu     =0        # (m)
ant2-postype     =rtcm    # (0:llh,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant2-pos1         =0        # (deg|m)
ant2-pos2         =0        # (deg|m)
ant2-pos3         =0        # (m|m)
ant2-anttype     =TRM29659.00
ant2-antdele     =0        # (m)
ant2-antdeln     =0        # (m)
ant2-antdelu     =0        # (m)
misc-timeinterp  =off      # (0:off,1:on)
misc-sbasatsel   =0        # (0:all)
misc-rnxopt1     =
misc-rnxopt2     =
file-satantfile =Y:\madoca\data\igs08.atx
file-rcvantfile  =Y:\madoca\data\igs08.atx
file-staposfile  =
file-geoidfile   =
file-ionofile    =
file-dcbfile     =Y:\madoca\data\dcb\P1P21201.DCB
file-eopfile     =
file-blqfile    =
file-tempdir     =C:\Temp
file-geexefile   =
file-solstatfile =
file-tracefile   =

inpstr1-type     =ntripcli #
(0:off,1:serial,2:file,3:tcpsvr,4:tcpcli,7:ntripcli,8:ftp,9:http)
inpstr2-type     =off      #
(0:off,1:serial,2:file,3:tcpsvr,4:tcpcli,7:ntripcli,8:ftp,9:http)
inpstr3-type     =off      #
(0:off,1:serial,2:file,3:tcpsvr,4:tcpcli,7:ntripcli,8:ftp,9:http)
inpstr1-path     =kaiyodai:tuomsat00@mgex.igs-ip.net:2101/CUT07:
inpstr2-path     =
inpstr3-path     =
inpstr1-format   =rtcm3   #
(0:rtcm2,1:rtcm3,2:oem4,3:oem3,4:ubx,5:ss2,6:hemis,7:skytraq,8:gw10,9:javad,15:sp3)
inpstr2-format   =rtcm3   #
(0:rtcm2,1:rtcm3,2:oem4,3:oem3,4:ubx,5:ss2,6:hemis,7:skytraq,8:gw10,9:javad,15:sp3)
inpstr3-format   =rtcm3   #
(0:rtcm2,1:rtcm3,2:oem4,3:oem3,4:ubx,5:ss2,6:hemis,7:skytraq,8:gw10,9:javad,15:sp3)

```

```
inpstr2-nmeareq      =off      # (0:off,1:latlon,2:single)
inpstr2-nmealat      =26.37293571 # (deg)
inpstr2-nmealon      =127.143649075 # (deg)
outstr1-type          =off      # (0:off,1:serial,2:file,3:tcpsvr,4:tcpcli,6:ntripsvr)
outstr2-type          =off      # (0:off,1:serial,2:file,3:tcpsvr,4:tcpcli,6:ntripsvr)
outstr1-path          =
outstr2-path          =
outstr1-format        =llh     # (0:llh,1:xyz,2:enu,3:nmea)
outstr2-format        =nmea   # (0:llh,1:xyz,2:enu,3:nmea)
logstr1-type          =off      # (0:off,1:serial,2:file,3:tcpsvr,4:tcpcli,6:ntripsvr)
logstr2-type          =off      # (0:off,1:serial,2:file,3:tcpsvr,4:tcpcli,6:ntripsvr)
logstr3-type          =off      # (0:off,1:serial,2:file,3:tcpsvr,4:tcpcli,6:ntripsvr)
logstr1-path          =
logstr2-path          =
logstr3-path          =
misc-svrcycle         =10       # (ms)
misc-timeout           =30000   # (ms)
misc-reconnect         =10000   # (ms)
misc-nmeacycle         =5000    # (ms)
misc-buffsize          =32768   # (bytes)
misc-navmsgsel         =all     # (0:all,1:rover,2:base,3:corr)
misc-proxyaddr         =
misc-fswapmargin       =30      # (s)
```

## B.5 URL List File for GNSS Data

### DESCRIPTION

A file containing URL list of GNSS data in Internet resources. The file is used by RTKGET to download GNSS data. A line indicates a record of URL for a GNSS data type. The strings after # are treated as comments. The following table shows the format of the URL list file for GNSS data.

No	Record/Field	Description	Notes
1	URL	A line contains a URL for a GNSS data type, which consists of the following fields separated by spaces.	
(1)	Data type	GNSS data type ID. The ID does not contain spaces. Under-bar "_" in the data type ID means the separator of each data type category.	
(2)	URL address	URL address of the GNSS data type. The URL address shall be:  <code>ftp://&lt;host address&gt;/&lt;file path&gt;</code> or  <code>http://&lt;host address&gt;/&lt;file path&gt;</code>  <code>ftp, http</code> : used download protocol <code>&lt;host address&gt;</code> : address of the host <code>&lt;file path&gt;</code> : download file path in the host. the file path can contain the following keywords replaced by date, time, station name and environment variable.  <pre>%Y -&gt; yyyy      : year (4 digits) (2000-2099) %y -&gt; yy       : year (2 digits) (00-99) %m -&gt; mm       : month          (01-12) %d -&gt; dd       : day of month   (01-31) %h -&gt; hh       : hours           (00-23) %H -&gt; a        : hour code       (a-x) %M -&gt; mm       : minutes          (00-59) %n -&gt; ddd      : day of year     (001-366) %W -&gt; wwww     : gps week        (0001-9999) %D -&gt; d        : day of gps week (0-6) %N -&gt; d        : sequence number (0- ) %s -&gt; ssss     : station name    (lower-case) %S -&gt; SSSS     : station name    (upper-case) %r -&gt; rrrr     : station name %{env} -&gt; env  : environment variable</pre>	
(3)	Default local directory	If the local directory is not specified in RTKGET, the downloaded files are saved in the directory. The directory path can contain keyword as same as URL address.	

**EXAMPLE**

```
#-----  
# data type url Address                               default local dir  
#-----  
  
#! PRODUCT (CDDIS)  
  
IGS_EPH ftp://cddis.gsfc.nasa.gov/gps/products/%W/igs%W%D.sp3.Z      c:\product\%W  
IGS_EPH_GLO ftp://cddis.gsfc.nasa.gov/gps/products/%W/igl%W%D.sp3.Z    c:\product\%W  
IGS_CLK ftp://cddis.gsfc.nasa.gov/gps/products/%W/igs%W%D.clk.Z       c:\product\%W  
IGS_CLK_30S ftp://cddis.gsfc.nasa.gov/gps/products/%W/igs%W%D.clk_30s.Z c:\product\%W  
IGS_ERP ftp://cddis.gsfc.nasa.gov/gps/products/%W/igs%W7.erp.Z        c:\product\%W  
IGR_EPH ftp://cddis.gsfc.nasa.gov/gps/products/%W/igr%W%D.sp3.Z       c:\product\%W  
IGR_CLK ftp://cddis.gsfc.nasa.gov/gps/products/%W/igr%W%D.clk.Z       c:\product\%W  
IGR_ERP ftp://cddis.gsfc.nasa.gov/gps/products/%W/igr%W%D.erp.Z       c:\product\%W  
IGU_EPH ftp://cddis.gsfc.nasa.gov/gps/products/%W/igu%W%D_%h.sp3.Z     c:\product\%W  
IGU_ERP ftp://cddis.gsfc.nasa.gov/gps/products/%W/igu%W%D_%h.erp.Z     c:\product\%W  
IGS_POS ftp://cddis.gsfc.nasa.gov/gps/products/%W/igs%yP%W.snx.Z       c:\product\%W  
IGS_TEC ftp://cddis.gsfc.nasa.gov/gps/products/ionex/%Y/%n/igsg%n0.%yi.Z c:\product\%W  
IGR_TEC ftp://cddis.gsfc.nasa.gov/gps/products/ionex/%Y/%n/igrg%n0.%yi.Z c:\product\%W  
...  
.
```

## Appendix C API References

The following table shows the list of RTKLIB library functions. For detailed API (calling convention, description of the function, input and output parameters and types, return value and type) for a library function, refer the header comment of each function in the source program in `<install dir>\rtklib_<ver>\src`. The definition of data types regarding to the APIs, refer the header file `rtklib.h` in `<install dir>\rtklib_<ver>\src`.

The detailed specifications and some examples of API usages will be also provided as RTKLIB API Reference Manual.

Table C-1 RTKLIB API function List (\*: added in ver. 2.4.2, \*\*: modified in ver. 2.4.2)

Function	Description	Source Program
Satellite number/system functions		
<code>satno()</code>	Satellite system and PRN/slot number to satellite number	<code>rtkcmn.c</code>
<code>satsys()</code>	Satellite number to satellite system	<code>rtkcmn.c</code>
<code>satid2no()</code>	Satellite ID to satellite number	<code>rtkcmn.c</code>
<code>satno2id()</code>	Satellite number to satellite ID	<code>rtkcmn.c</code>
<code>obs2code()</code>	Observation type string to observation code	<code>rtkcmn.c **</code>
<code>code2obs()</code>	Observation code to observation code string	<code>rtkcmn.c **</code>
<code>satexclude()</code>	Test excluded satellites	<code>rtkcmn.c *</code>
<code>testsnr()</code>	Test SNR mask	<code>rtkcmn.c *</code>
<code>setcodepri()</code>	Set code priority for multiple codes	<code>rtkcmn.c *</code>
<code>getcodepri()</code>	Get code priority for multiple codes	<code>rtkcmn.c *</code>
Matrix and vector functions		
<code>mat()</code>	New matrix	<code>rtkcmn.c</code>
<code>imat()</code>	New integer matrix	<code>rtkcmn.c</code>
<code>zeros()</code>	New zero matrix	<code>rtkcmn.c</code>
<code>eye()</code>	New identity matrix	<code>rtkcmn.c</code>
<code>dot()</code>	Inner Product	<code>rtkcmn.c</code>
<code>norm()</code>	Euclid norm	<code>rtkcmn.c</code>
<code>cross3()</code>	Outer product of 3D vectors	<code>rtkcmn.c</code>
<code>normv3()</code>	Normalize 3D vector	<code>rtkcmn.c</code>
<code>matcpy()</code>	Copy matrix	<code>rtkcmn.c</code>
<code>matmul()</code>	Multiply matrix	<code>rtkcmn.c</code>
<code>matinv()</code>	Inverse of matrix	<code>rtkcmn.c</code>
<code>solve()</code>	Solve linear equation	<code>rtkcmn.c</code>
<code>lsq()</code>	Least square estimation	<code>rtkcmn.c</code>
<code>filter()</code>	Kalman filter state update	<code>rtkcmn.c</code>
<code>smoother()</code>	Kalman smoother	<code>rtkcmn.c</code>
<code>matprint()</code>	Print matrix	<code>rtkcmn.c</code>
<code>matfprint()</code>	Print matrix to file	<code>rtkcmn.c</code>
Time and string functions		

Function	Description	Source Program
<b>str2num()</b>	String to number	<b>rtkcmn.c</b>
<b>str2time()</b>	String to time	<b>rtkcmn.c</b>
<b>time2str()</b>	Time to string	<b>rtkcmn.c</b>
<b>epoch2time()</b>	Calendar day/time to time	<b>rtkcmn.c</b>
<b>time2epoch()</b>	Time to calendar day/time	<b>rtkcmn.c</b>
<b>gpst2time()</b>	GPS week/TOW to time	<b>rtkcmn.c</b>
<b>time2gpst()</b>	Time to GPS week/TOW	<b>rtkcmn.c</b>
<b>gst2time()</b>	GST week/TOW to time	<b>rtkcmn.c</b>
<b>time2gst()</b>	Time to GST week/TOW	<b>rtkcmn.c</b>
<b>bdt2time()</b>	BDT week/TOW to time	<b>rtkcmn.c *</b>
<b>time2bdt()</b>	Time to BDT week/TOW	<b>rtkcmn.c *</b>
<b>time_str()</b>	Get Time String	<b>rtkcmn.c</b>
<b>timeadd()</b>	Add time	<b>rtkcmn.c</b>
<b>timediff()</b>	Time difference	<b>rtkcmn.c</b>
<b>gpst2utc()</b>	GPS Time to UTC	<b>rtkcmn.c</b>
<b>utc2gpst()</b>	UTC to GPS Time	<b>rtkcmn.c</b>
<b>gpst2bdt()</b>	GPS Time to BDT	<b>rtkcmn.c *</b>
<b>bdt2gpst()</b>	BDT to GPS Time	<b>rtkcmn.c *</b>
<b>timeget()</b>	Get current time in UTC	<b>rtkcmn.c</b>
<b>timeset()</b>	Set current time in UTC	<b>rtkcmn.c</b>
<b>time2doy()</b>	Time to Day of Year	<b>rtkcmn.c</b>
<b>utc2gmst()</b>	UTC to Greenwich Mean Sidereal Time	<b>rtkcmn.c *</b>
<b>adjgpsweek()</b>	Adjust GPS week number	<b>rtkcmn.c</b>
<b>tickget()</b>	Get current tick time	<b>rtkcmn.c</b>
<b>sleepms()</b>	Sleep for milliseconds	<b>rtkcmn.c</b>
<b>reppath()</b>	Replace file path	<b>rtkcmn.c</b>
<b>reppaths()</b>	Replace file paths	<b>rtkcmn.c</b>
Coordinates functions		
<b>ecef2pos()</b>	ECEF to geodetic position	<b>rtkcmn.c</b>
<b>pos2ecef()</b>	Geodetic to ECEF position	<b>rtkcmn.c</b>
<b>ecef2enu()</b>	ECEF to local coordinates	<b>rtkcmn.c</b>
<b>enu2ecef()</b>	Local to ECEF coordinates	<b>rtkcmn.c</b>
<b>covenu()</b>	Covariance in local coordinates	<b>rtkcmn.c</b>
<b>covecef()</b>	Covariance in ECEF coordinates	<b>rtkcmn.c</b>
<b>xyz2enu()</b>	ECEF to ENU local coordinate transformation matrix	<b>rtkcmn.c</b>
<b>eci2ecef()</b>	ECI to ECEF transformation matrix	<b>rtkcmn.c **</b>
<b>deg2dms()</b>	Convert degree to deg-min-sec	<b>rtkcmn.c</b>
<b>dms2deg()</b>	Convert deg-min-sec to degree	<b>rtkcmn.c</b>
Input/output functions		
<b>readpos()</b>	Read station positions	<b>rtkcmn.c</b>
<b>sortobs()</b>	Sort observation data	<b>rtkcmn.c</b>
<b>uniqnav()</b>	Delete duplicated ephemerides in navigation data	<b>rtkcmn.c</b>
<b>screen()</b>	Screen data by time and interval	<b>rtkcmn.c</b>
<b>readnav()</b>	Read navigation data from file	<b>rtkcmn.c</b>
<b>savenav()</b>	Save navigation data to file	<b>rtkcmn.c</b>
<b>freeobs()</b>	Free memory for observation data	<b>rtkcmn.c</b>
<b>freenav()</b>	Free memory for navigation data	<b>rtkcmn.c</b>
Debug trace functions		
<b>traceopen()</b>	Open trace file	<b>rtkcmn.c</b>

Function	Description	Source Program
<b>traceclose()</b>	Close trace file	<b>rtkcmn.c</b>
<b>trace()</b>	Output trace	<b>rtkcmn.c</b>
<b>tracet()</b>	Output trace with time tag	<b>rtkcmn.c</b>
<b>tracemat()</b>	Output trace as matrix printing	<b>rtkcmn.c</b>
<b>traceobs()</b>	Output trace as observation data printing	<b>rtkcmn.c</b>
<b>traceonav()</b>	Output trace as GPS navigation messages	<b>rtkcmn.c</b>
<b>tracegnav()</b>	Output trace as GLONASS navigation messages	<b>rtkcmn.c</b>
<b>tracehnnav()</b>	Output trace as GEO navigation messages	<b>rtkcmn.c</b>
<b>tracepeph()</b>	Output trace as precise ephemeris	<b>rtkcmn.c</b>
<b>tracepclk()</b>	Output trace as precise clock	<b>rtkcmn.c</b>
<b>traceb()</b>	Output trace as binary dump	<b>rtkcmn.c</b>
Platform dependent functions		
<b>execcmd()</b>	Execute command	<b>rtkcmn.c</b>
<b>expath()</b>	Expand file path	<b>rtkcmn.c</b>
<b>createdir()</b>	Create directory	<b>rtkcmn.c</b>
Positioning models		
<b>satwavelen()</b>	Satellite signal carrier wave length	<b>rtkcmn.c</b>
<b>satazel()</b>	Satellite azimuth/elevation angle	<b>rtkcmn.c</b>
<b>geodist()</b>	Geometric distance	<b>rtkcmn.c</b>
<b>dops()</b>	Compute DOPs	<b>rtkcmn.c</b>
<b>csmooth()</b>	Carrier smoothing	<b>rtkcmn.c</b>
Atmosphere models		
<b>ionmodel()</b>	Ionospheric model	<b>rtkcmn.c</b>
<b>ionmapf()</b>	Ionospheric mapping function	<b>rtkcmn.c</b>
<b>ionppp()</b>	Ionospheric pierce point position	<b>rtkcmn.c</b>
<b>tropmodel()</b>	Tropospheric model	<b>rtkcmn.c</b>
<b>tropmapf()</b>	Tropospheric mapping function	<b>rtkcmn.c</b>
<b>iontec()</b>	Ionosphere model by TEC grid data	<b>ionex.c</b>
<b>readtec()</b>	Read IONEX TEC grid file	<b>ionex.c</b>
<b>ionocorr()</b>	Ionosphere correction	<b>pntpos.c</b>
<b>tropcorr()</b>	Troposphere correction	<b>pntpos.c</b>
Antenna models		
<b>readpcv()</b>	Read antenna parameters	<b>rtkcmn.c</b>
<b>searchpcv()</b>	Search antenna parameters	<b>rtkcmn.c</b>
<b>antmodel()</b>	Receiver antenna model	<b>rtkcmn.c **</b>
<b>antmodel_s()</b>	Satellite antenna model	<b>rtkcmn.c</b>
Earth tides models		
<b>sunmoonpos()</b>	Sun and moon position	<b>rtkcmn.c **</b>
<b>tidedisp()</b>	Displacements by earth tides	<b>PPP.c</b>
Geoid model		
<b>opengeoid()</b>	Open external geoid file	<b>geoid.c</b>
<b>closegeoid()</b>	Close external geoid file	<b>geoid.c</b>
<b>geoidh()</b>	Geoid height	<b>geoid.c</b>
Datum transformation		
<b>loaddatump()</b>	Load datum transformation parameter	<b>datum.c</b>
<b>tokyo2jgd()</b>	Tokyo datum to JGD2000 datum	<b>datum.c</b>

Function	Description	Source Program
<code>jgd2tokyo()</code>	JGD2000 datum to Tokyo datum	<code>datum.c</code>
RINEX functions		
<code>readrnx()</code>	Read RINEX file	<code>rinex.c **</code>
<code>readrnxt()</code>	Read RINEX file in time range/interval	<code>rinex.c **</code>
<code>readrnxc()</code>	Read RINEX clock file	<code>rinex.c</code>
<code>outrnxobsh()</code>	Output RINEX OBS header	<code>rinex.c **</code>
<code>outrnxobsb()</code>	Output RINEX OBS body	<code>rinex.c</code>
<code>outrnxnavh()</code>	Output RINEX NAV header	<code>rinex.c</code>
<code>outrnxgnavh()</code>	Output RINEX GLONASS NAV header	<code>rinex.c</code>
<code>outrnxhnnavh()</code>	Output RINEX GEO NAV header	<code>rinex.c</code>
<code>outrnxlnavh()</code>	Output RINEX Galileo NAV header	<code>rinex.c</code>
<code>outrnxqnavh()</code>	Output RINEX QZSS NAV header	<code>rinex.c</code>
<code>outrnxcnavh()</code>	Output RINEX BeiDou NAV header	<code>rinex.c *</code>
<code>outrnxnavb()</code>	Output RINEX NAV body	<code>rinex.c</code>
<code>outrnxgnavb()</code>	Output RINEX GLONASS NAV body	<code>rinex.c</code>
<code>outrnxhnnavb()</code>	Output RINEX GEO NAV body	<code>rinex.c</code>
<code>uncompress()</code>	Uncompress file	<code>rinex.c</code>
<code>init_rnxctr()</code>	Initialize RINEX control	<code>rinex.c *</code>
<code>free_rnxctr()</code>	Free RINEX control	<code>rinex.c *</code>
<code>open_rnxctr()</code>	Open RINEX control	<code>rinex.c *</code>
<code>input_rnxctr()</code>	Input next RINEX data by RINEX control	<code>rinex.c *</code>
<code>convrnx()</code>	RINEX converter	<code>convrnx.c</code>
Ephemeris functions		
<code>eph2clk()</code>	Broadcast ephemeris to satellite clock-bias	<code>ephemeris.c</code>
<code>geph2clk()</code>	GLONASS ephemeris to satellite clock-bias	<code>ephemeris.c</code>
<code>seph2clk()</code>	GEO ephemeris to satellite clock-bias	<code>ephemeris.c</code>
<code>eph2pos()</code>	Broadcast ephemeris to satellite position/clock-bias	<code>ephemeris.c</code>
<code>geph2pos()</code>	GLONASS ephemeris to satellite position/clock-bias	<code>ephemeris.c</code>
<code>seph2pos()</code>	GEO ephemeris to satellite position/clock-bias	<code>ephemeris.c</code>
<code>peph2pos()</code>	Precise ephemeris to satellite position/clock-bias	<code>preceph.c</code>
<code>satantoff()</code>	Satellite antenna phase center offset	<code>ephemeris.c</code>
<code>satpos()</code>	Satellite position/clock-bias	<code>ephemeris.c</code>
<code>satposs()</code>	Satellite positions/clock-biases	<code>ephemeris.c</code>
<code>readsp3()</code>	Read SP3 file	<code>preceph.c **</code>
<code>readsap()</code>	Read satellite antenna phase center parameters	<code>preceph.c</code>
<code>readdcb()</code>	Read DCB parameters	<code>preceph.c</code>
<code>alm2pos()</code>	Almanac to satellite position/clock-bias	<code>preceph.c</code>
<code>tle_read()</code>	Read TLE data file	<code>tle.c *</code>
<code>tle_name_read()</code>	Read TLE satellite name file	<code>tle.c *</code>
<code>tle_pos()</code>	Satellite position and velocity with TLE data	<code>tle.c *</code>
Receiver raw data functions		
<code>getbitu()</code>	Extract unsigned bits	<code>rtkcmn.c</code>
<code>getbits()</code>	Extract signed bits	<code>rtkcmn.c</code>
<code>setbitu()</code>	Set unsigned bits	<code>rtkcmn.c *</code>
<code>setbits()</code>	Set signed bits	<code>rtkcmn.c *</code>
<code>crc32()</code>	CRC32 parity	<code>rtkcmn.c</code>
<code>crc24q()</code>	CRC24Q parity	<code>rtkcmn.c</code>
<code>crc16()</code>	CRC16 parity	<code>rtkcmn.c *</code>
<code>decode_word()</code>	Decode navigation data word	<code>rcvraw.c</code>

Function	Description	Source Program
<code>decode_frame()</code>	Decode navigation data frame	<code>rcvraw.c</code>
<code>init_raw()</code>	Initialize receiver raw data control	<code>rcvraw.c</code>
<code>free_raw()</code>	Free receiver raw data control	<code>rcvraw.c</code>
<code>input_raw()</code>	Input receiver raw data from stream	<code>rcvraw.c</code>
<code>input_rawf()</code>	Input receiver raw data from file	<code>rcvraw.c</code>
Receiver dependent functions		
<code>input_oem4()</code>	Input OEM4/V raw data from stream	<code>rcv/novatel.c</code>
<code>input_oem3()</code>	Input OEM3 raw data from stream	<code>rcv/novatel.c</code>
<code>input_ubx()</code>	Input u-blox raw data from stream	<code>rcv/ublox.c</code>
<code>input_ss2()</code>	Input Superstar II raw data from stream	<code>rcv/ss2.c</code>
<code>input_cres()</code>	Input Crescent raw data from stream	<code>rcv/crescent.c</code>
<code>input_stq()</code>	Input SkyTraq raw data from stream	<code>rcv/skytraq.c</code>
<code>input_gw10()</code>	Input Furuno GW-10-II/III raw data from stream	<code>rcv/gw10.c</code>
<code>input_javad()</code>	Input JAVAD GRIL/GREIS raw data from stream	<code>rcv/javad.c</code>
<code>input_nvs()</code>	Input NVS BINR raw data from stream	<code>rcv/nvs.c *</code>
<code>input_binex()</code>	Input BINEX data from stream	<code>rcv/binex.c *</code>
<code>input_oem3f()</code>	Input OEM3 raw data from file	<code>rcv/novatel.c</code>
<code>input_oem4f()</code>	Input OEM4/V raw data from file	<code>rcv/novatel.c</code>
<code>input_ubxf()</code>	Input u-blox raw data from file	<code>rcv/ublox.c</code>
<code>input_ss2f()</code>	Input Superstar II raw data from file	<code>rcv/ss2.c</code>
<code>input_cresf()</code>	Input Crescent raw data from file	<code>rcv/crescent.c</code>
<code>input_stqf()</code>	Input SkyTraq raw data from file	<code>rcv/skytraq.c</code>
<code>input_gw10f()</code>	Input Furuno GW-10-II/III raw data from file	<code>rcv/gw10.c</code>
<code>input_javadf()</code>	Input JAVAD GRIL/GREIS raw data from file	<code>rcv/javad.c</code>
<code>input_nvsf()</code>	Input NVS BINR raw data from file	<code>rcv/nvs.c *</code>
<code>input_binexf()</code>	Input BINEX data from file	<code>rcv/binex.c *</code>
<code>gen_ubx()</code>	Generate u-blox binary command	<code>rcv/ublox.c</code>
<code>gen_stq()</code>	Generate SkyTraq binary command	<code>rcv/skytraq.c</code>
<code>gen_nvs()</code>	Generate NVS BINR binary command	<code>rcv/nvs.c *</code>
RTCM functions		
<code>init_rtcm()</code>	Initialize RTCM control	<code>rtcm.c</code>
<code>free_rtcm()</code>	Free RTCM control	<code>rtcm.c</code>
<code>input_rtcm2()</code>	Input RTCM 2 message from stream	<code>rtcm.c</code>
<code>input_rtcm3()</code>	Input RTCM 3 message from stream	<code>rtcm.c</code>
<code>input_rtcm2f()</code>	Input RTCM 2 message from file	<code>rtcm.c</code>
<code>input_rtcm3f()</code>	Input RTCM 3 message from file	<code>rtcm.c</code>
<code>gen_rtcm2()</code>	Generate RTCM 2 message	<code>rtcm.c *</code>
<code>gen_rtcm3()</code>	Generate RTCM 3 message	<code>rtcm.c *</code>
Solution functions		
<code>initsolbuf()</code>	Initialize solution buffer	<code>solution.c</code>
<code>freesolbuf()</code>	Free solution buffer	<code>solution.c</code>
<code>freesolstatbuf()</code>	Free solution status buffer	<code>solution.c</code>
<code>getsol()</code>	Get solution data from solution buffer	<code>solution.c</code>
<code>addsol()</code>	Add solution data to solution buffer	<code>solution.c</code>
<code>readsol()</code>	Read solution data from solutions files	<code>solution.c</code>
<code>readsolt()</code>	Read solution data in time range/interval	<code>solution.c</code>
<code>readsolstat()</code>	Read solution status from file	<code>solution.c</code>
<code>readsolstatt()</code>	Read solution status in time range/interval	<code>solution.c</code>
<code>inputsol()</code>	Input solution data from stream	<code>solution.c</code>

Function	Description	Source Program
<code>outprcopts()</code>	Output processing options to string	<code>solution.c</code>
<code>outsolheads()</code>	Output solution header to string	<code>solution.c</code>
<code>outsols()</code>	Output solution body to string	<code>solution.c</code>
<code>outsolexs()</code>	Output extended solution to string	<code>solution.c</code>
<code>outprcopt()</code>	Output processing options to file	<code>solution.c</code>
<code>outsolhead()</code>	Output solution header to file	<code>solution.c</code>
<code>outsol()</code>	Output solution body to file	<code>solution.c</code>
<code>outsolex()</code>	Output extended solution to file	<code>solution.c</code>
<code>outnmea_rmc()</code>	Output NMEA GPRMC sentence	<code>solution.c</code>
<code>outnmea_gga()</code>	Output NMEA GPGGA sentence	<code>solution.c</code>
<code>outnmea_gsa()</code>	Output NMEA GPGSA, GLGSA, GAGSA sentences	<code>solution.c</code>
<code>outnmea_gsv()</code>	Output NMEA GPGSV, GLGSV, GAGSV sentences	<code>solution.c</code>
Convert solutions to Google Earth KML file		
<code>convkml()</code>	Convert solutions to Google Earth KML file	<code>convkml.c</code>
SBAS functions		
<code>sbsreadmsg()</code>	Read SBAS message file	<code>sbas.c</code>
<code>sbsreadsmgt()</code>	Read SBAS message file in time range	<code>sbas.c</code>
<code>sbsoutmsg()</code>	Output SBAS messages	<code>sbas.c</code>
<code>sbsdecodemsg()</code>	Decode SBAS message	<code>sbas.c</code>
<code>sbsupdatecorr()</code>	Update SBAS corrections	<code>sbas.c</code>
<code>sbssatcorr()</code>	SBAS satellite correction	<code>sbas.c</code>
<code>sbsioncorr()</code>	SBAS ionospheric correction	<code>sbas.c</code>
<code>sbstropcorr()</code>	SBAS tropospheric correction	<code>sbas.c</code>
Options functions		
<code>searchopt()</code>	Search option	<code>options.c</code>
<code>str2opt()</code>	String to option value	<code>options.c</code>
<code>opt2str()</code>	Option value to string	<code>options.c</code>
<code>opt2buf()</code>	Option to string	<code>options.c</code>
<code>loadopts()</code>	Load options from file	<code>options.c</code>
<code>saveopts()</code>	Save options to file	<code>options.c</code>
<code>resetsysopts()</code>	Reset system options to default	<code>options.c</code>
<code>getsysopts()</code>	Get system options	<code>options.c</code>
<code>setsysopts()</code>	Set system options	<code>options.c</code>
Stream data input/output functions		
<code>strinitcom()</code>	Initialize stream communication environment	<code>stream.c</code>
<code>strinit()</code>	Initialize stream	<code>stream.c</code>
<code>strlock()</code>	Lock stream	<code>stream.c</code>
<code>strunlock()</code>	Unlock stream	<code>stream.c</code>
<code>stropen()</code>	Open stream	<code>stream.c</code>
<code>strclose()</code>	Close stream	<code>stream.c</code>
<code>strread()</code>	Read stream	<code>stream.c</code>
<code>strwrite()</code>	Write stream	<code>stream.c</code>
<code>strsync()</code>	Time sync stream	<code>stream.c</code>
<code>strstat()</code>	Get stream status	<code>stream.c</code>
<code>strsum()</code>	Get stream statistics summary	<code>stream.c</code>
<code>strsetopt()</code>	Set stream options	<code>stream.c</code>
<code>str_gettime()</code>	Get current time from stream	<code>stream.c</code>
<code>strsendnmea()</code>	Send NMEA message to stream	<code>stream.c</code>

Function	Description	Source Program
<b>strsendcmd()</b>	Send receiver command to stream	<b>stream.c</b>
<b>strsettimeout()</b>	Set stream timeout parameters	<b>stream.c</b>
<b>strsetdir()</b>	Set local directory	<b>stream.c</b>
<b>strsetproxy()</b>	Set proxy address	<b>stream.c</b>
Integer ambiguity resolution		
<b>lambda()</b>	LAMBDA/MLAMBDA integer least-square estimation	<b>lambda.c</b>
Standard positioning		
<b>pntpos()</b>	Standard positioning	<b>pntpos.c</b>
Precise positioning		
<b>rtkinit()</b>	Initialize RTK control struct	<b>rtkpos.c</b>
<b>rtkfreet()</b>	Free RTK control struct	<b>rtkpos.c</b>
<b>rtkpos()</b>	Precise positioning	<b>rtkpos.c</b>
<b>rtkopenstat()</b>	Open solution status file	<b>rtkpos.c</b>
<b>rtkclosestat()</b>	Close solution status file	<b>rtkpos.c</b>
Precise point positioning		
<b>pppos()</b>	Precise point positioning (PPP)	<b>ppp.c *</b>
<b>pppnx()</b>	Number of estimated states for PPP	<b>ppp.c *</b>
<b>pppoutsolstat()</b>	Output solution statistics for PPP	<b>ppp.c *</b>
<b>windupcorr()</b>	Phase windup correction	<b>rtkcnn.c *</b>
<b>pppamb()</b>	Resolve integer ambiguity for PPP	<b>ppp_ar.c *</b>
Post-processing positioning		
<b>postpos()</b>	Post-processing positioning	<b>postpos.c</b>
Stream server functions		
<b>strsvrinit()</b>	Initialize stream server	<b>streamsrv.c</b>
<b>strsvrstart()</b>	Start stream server	<b>streamsrv.c **</b>
<b>strsvrstop()</b>	Stop stream server	<b>streamsrv.c</b>
<b>strsvrstat()</b>	Get stream server status	<b>streamsrv.c</b>
<b>strconvnew()</b>	Generate stream converter	<b>streamsrv.c *</b>
<b>strconvfree()</b>	Free stream converter	<b>streamsrv.c *</b>
RTK server functions		
<b>rtsvrvinit()</b>	Initialize RTK server	<b>rtksvr.c</b>
<b>rtsvrvstart()</b>	Start RTK server	<b>rtksvr.c</b>
<b>rtsvrvstop()</b>	Stop RTK server	<b>rtksvr.c</b>
<b>rtsvrvopenstr()</b>	Open output/log stream	<b>rtksvr.c</b>
<b>rtsvrvclosestr()</b>	Close output/log stream	<b>rtksvr.c</b>
<b>rtsvrvlock()</b>	Lock RTK server	<b>rtksvr.c</b>
<b>rtsvrvunlock()</b>	Unlock RTK server	<b>rtksvr.c</b>
<b>rtsvrvrostat()</b>	Get RTK observation data status	<b>rtksvr.c</b>
<b>rtsvrvssstat()</b>	Get RTK stream status	<b>rtksvr.c</b>
Downloader functions		
<b>dl_readurls()</b>	Read URL address list file of GNSS data	<b>download.c *</b>
<b>dl_readstas()</b>	Read station list file for download	<b>download.c *</b>
<b>dl_exec()</b>	Execute download of GNSS data	<b>download.c *</b>
<b>dl_test()</b>	Execute local file test of GNSS data	<b>download.c *</b>

Function	Description	Source Program
QZSS LEX functions		
<b>lexupdatecorr()</b>	Update LEX corrections	<b>qzslex.c *</b>
<b>lexreadmsg()</b>	Read LEX message log file	<b>qzslex.c *</b>
<b>lexoutmsg()</b>	Output LEX message log	<b>qzslex.c *</b>
<b>lexconvbin()</b>	Convert LEX binary to LEX message log	<b>qzslex.c *</b>
<b>lexeph2pos()</b>	LEX satellite ephemeris and clock correction	<b>qzslex.c *</b>
<b>lexioncorr()</b>	LEX ionosphere correction	<b>qzslex.c *</b>

## Appendix D Files and Messages

### D.1 Supported RINEX Files

Supported RINEX versions and files by RTKLIB are shown in the following table.

RINEX Version	Observation Data						Navigation Messages						Met.	CLK	GEO BRDC
	G	R	E	J	C	S	G	R	E	J	C	S			
2.10	O	O	O*	O*	O*	O	N	G	N*	N*	-	H	-	-	-
2.11	O	O	O	O*	O*	O	N	G	N*	N*	-	H	-	-	-
2.12	O	O	O	O*	O	O	N	G	N	N*	-	H	-	-	-
3.00	O	O	O	O*	O*	O	N	N	N	N*	N*	N	-	C**	-
3.01	O	O	O	O*	O	O	N	N	N	N*	N*	N	-	C**	-
3.02	O	O	O	O	O	O	N	N	N	N	N	N	-	C**	-

G: GPS, R: GLONASS, E: Galileo, J: QZSS, C: BeiDou, S: SBAS

-: not supported, O, N, G, H: supported as RINEX file type

\* RTKLIB extensions (QZSS extensions are based on JAXA [60][61]), \*\* read only,

## D.2 Supported Receiver Messages

Supported RTCM 2, RTCM 3, BINEX and receivers' proprietary messages by RTKLIB are shown in the following table.

Format	Data Message Types					
	Raw Observation Data	Satellite Ephemerides	ION/UTC Parameters	Antenna Info	SBAS Messages	Others
RTCM 2 [16]	18, 19	17	-	3, 22	-	1*, 9*, 14, 16
RTCM 3 [17][18]	see below	see below	-	see below	-	see below
BINEX [19] **	0x7f-05 (Trimble NetR8)	0x01-01, 0x01-02, 0x01-03, 0x01-04, 0x01-06	-	-	-	-
NovAtel OEM4/V/6 [41][42]	RANGEB, RANGECMPB	RAWEPEMB, GLO- EPHEMERISB, QZSS- RAWEPEMB, GAL- EPHEMERISB	IONUTCB, QZSS- IONUTCB, GALIONOB, GALCLOCKB	-	RAWWAAS- FRAMEB, RAWSBAS- FRAMEB, QZSSRAW- SUBFRAMEB	-
NovAtel OEM3 [43]	RGBE, RGED	REPB	IONB, UTCB	-	FRMB	-
u-blox LEA-4T/5T/ 6T/8/9 [44]	UBX RXM-RAW RXM-RAWX	UBX RXM-SFRB	UBX RXM-SFRB	-	UBX RXM-SFRB	-
NovAtel Superstar II [45]	ID#23	ID#22	-	-	ID#67	ID#20, ID#21
Hemisphere Crescent, Eclipse [46][47]	bin 96, bin 76	bin 95	bin 94	-	bin 80	-
SkyTraq S1315F [48][49]	msg 0xDD (221)	msg 0xE0 (224)	msg 0xE0 (224)	-	-	msg 0xDC (220)
Furuno GW-10-II/III [50]	msg 0x08	msg 0x27	msg 0x27	-	msg 0x27	msg 0x20
JAVAD GRIL/GRIES [51][52][53][54]	[RC], [rc], [CR], [cr], [PC], [pc], [CP], [cp], [DC], [cd], [EC], [CE], [FC], [R1], [r1], [1R], [1r], [P1], [p1], [1P], [1p], [D1], [1d], [E1], [1E], [F1], [R2],	[GE], [NE], [EN], [WE], [QE]	[UO], [NU], [EU], [WU], [QU], [IO]	-	[WD]	[~~], [::], [RD], [ST], [NN]

Format	Data Message Types					
	Raw Observation Data	Satellite Ephemerides	ION/UTC Parameters	Antenna Info	SBAS Messages	Others
[r2], [2R], [2r], [P2], [p2], [2P], [2p], [D2], [2d], [E2], [2E], [F2], [R3], [r3], [3R], [3r], [P3], [p3], [3P], [3p], [D3], [3d], [E3], [3E], [F3], [R5], [r5], [5R], [5r], [P5], [p5], [5P], [5p], [D5], [5d], [E5], [5E], [F5], [R1], [r1], [1R], [1r], [P1], [p1], [1P], [1p], [D1], [1d], [E1], [1E], [F1], [TC]						
NVS NV08C [55][56]	msg F5h	msg F7h	msg 4Ah, msg 4Bh	-	-	-

\* Only support to read, DGPS correction is not supported

\*\* Only big-endian, forward and regular CRC messages

## Supported RTCM 3 Message Types

Message	GPS	GLOASS	Galileo	QZSS	BeiDou	SBAS
OBS Compact L1	1001~	1009~	-	-	-	-
Full L1	1002	1010	-	-	-	-
Compact L1/2	1003~	1011~	-	-	-	-
Full L1/2	1004	1012	-	-	-	-
Ephemeris	1019	1020	1045*	1044*	-	-
	-	-	1046*	-	-	-
MSM 1	1071~	1081~	1091~	1111*~	1121*~	1101*~
2	1072~	1082~	1092~	1112*~	1122*~	1102*~
3	1073~	1083~	1093~	1113*~	1123*~	1103*~
4	1074	1084	1094	1114*	1124*	1104*
5	1075	1085	1095	1115*	1125*	1105*
6	1076	1086	1096	1116*	1126*	1106*
7	1077	1087	1097	1117*	1127*	1107*
SSR Orbit Corr.	1057	1063	1240*	1246*	-	-
Clock Corr.	1058	1064	1241*	1247*	-	-
Code Bias	1059	1065	1242*	1248*	-	-
Combined	1060	1066	1243*	1249*	-	-
URA	1061	1067	1244*	1250*	-	-
HR-Clock	1062	1068	1245*	1251*	-	-
Antenna Info	1005	1006	1007	1008	1033	

\* draft, ~ only encode

### D.3 Supported Signal IDs/Observation Types

Supported signal IDs/observation types by RTKLIB are shown in the following table. The table also indicate the correspondent RINEX 2 and RINEX 3 observation types, RTCM 3 MSM signal IDs and BINEX observation code IDs. For RTCM 2, RTCM3 and RINEX to RINEX conversion by RTKCONV and CONVBIN, RTCM 2 and BINEX to RTCM 3 conversion by STRSVR and STR2STR, the table is used as well.

System	Freq.	Channel or Code	Signal ID	RINEX 2 *1	RINEX 3 *2	RTCM 3 *3	BINEX *4
GPS	L1	C/A	1C	C1/CA*	1C	2	0,1
		L1C(D)	1S	-	1S	30	-
		L1C(P)	1L	-	1L	31	-
		L1C(D+P)	1X	CB*	1X	32	6
		P	1P	-	1P	3	2
		Z-tracking and similar (AS on)	1W	P1/C1*	1W	4	3
		Y	1Y	-	1Y	5	4
		M	1M	-	1M	6	5
		codeless	1N	-	1N	-	7
		C/A	2C	-	2C	8	11
	L2	L1(C/A)+(P2-P1) (semi-codeless)	2D	-	2D	-	12
		L2C(M)	2S	-	2S	15	13
		L2C(L)	2L	-	2L	16	14
		L2C(M+L)	2X	C2/CC*	2X	17	15
		P	2P	-	2P	9	16
GLONASS	L5	Z-tracking and similar (AS on)	2W	P2	2W	10	10,17
		Y	2Y	-	2Y	11	18
		M	2M	-	2M	12	19
		codeless	2N	-	2N	-	20
		I	5I	-	5I	22	24
		Q	5Q	-	5Q	23	25
		I+Q	5X	C5	5X	24	23,26
	G1	C/A	1C	C1/CA*	1C	2	0,1
		P	1P	P1/C1*	1P	3	2
Galileo	G2	C/A	2C	C2/CD*	2C	8	10,11
		P	2P	P2	2P	9	12
		I	3I	-	3I	11	14
	G3	Q	3Q	-	3Q	12	15
		I+Q	3X	-	3X	13	13,16
	E1	A PRS	1A	-	1A	3	1
		B I/NAV OS/CS/SoL	1B	-	1B	4	2
		C	1C	-	1C	2	0,3
		B+C	1X	C1	1X	5	4
		A+B+C	1Z	-	1Z	6	5
	E5a	I F/NAV OS	5I	-	5I	22	7

System	Freq.	Channel or Code	Signal ID	RINEX 2 *1	RINEX 3 *2	RTCM 3 *3	BINEX *4	
E5b		Q no data	5Q	-	5Q	23	8	
		I+Q	5X	C5	5X	24	6, 9	
		I I/NAV OS/CS/SoL	7I	-	7I	14	11	
		Q no data	7Q	-	7Q	15	12	
		I+Q	7X	C7	7X	16	10, 13	
		I	8I	-	8I	18	15	
		Q	8Q	-	8Q	19	16	
		I+Q	8X	C8	8X	20	14, 17	
		A PRS	6A	-	6A	9	19	
		B C/NAV CS	6B	-	6B	10	20	
E6		C no data	6C	-	6C	8	21	
		B+C	6X	C6	6X	11	18, 22	
		A+B+C	6Z	-	6Z	12	23	
		C/A	1C	C1/CA*	1C	2	0, 1	
		L1C(D)	1S	-	1S	30	2	
QZSS	L1	L1C(P)	1L	-	1L	31	3	
		L1C(D+P)	1X	CB*	1X	32	4	
		L1-SAIF	1Z	-	1Z	6	30	
		L2C(M)	2S	-	2S	15	8	
		L2C(L)	2L	-	2L	16	9	
	L2	L2C(M+L)	2X	C2/CC*	2X	17	7, 10	
		I	5I	-	5I	22	14	
		Q	5Q	-	5Q	23	15	
		I+Q	5X	-	5X	24	13, 16	
		S	6S	-	6S	9	20	
BeiDou	LEX	L	6L	-	6L	10	21	
		S+L	6X	C6	6X	11	19, 22	
		I	2I	-	2I	2	1	
		Q	2Q	-	2Q	3	2	
		I+Q	2X	C2	2X	4	0, 3	
	B1	I	7I	-	7I	14	5	
		Q	7Q	-	7Q	15	6	
		I+Q	7X	C7	7X	16	4, 7	
		I	6I	-	6I	8	9	
		Q	6Q	-	6Q	9	10	
SBAS	B3	I+Q	6X	C6	6X	10	8, 11	
		L1	C/A	1C	C1/CA*	1C	2	0, 1
		I	5I	-	5I	22	7	
		Q	5Q	-	5Q	23	8	
		I+Q	5X	C5	5X	24	6, 9	

\*1 pseudorange OBS TYPE [9][10][11][12][13][14], \*2 [12][13][14], \*3 RTCM MSM signal ID [18],

\*4 BINEX observation code ID for message 0x7f-05 [19], \* RINEX 2.12

## D.4 Default Priorities of Multiple Signals

If input observation data contain multiple signals in a frequency, RTKLIB selects a signal for processing by the following default priorities. To select appropriate signal, use RINEX options or receiver dependent options described in Appendix D.5. In user APs, to change or obtain the signal priorities, use API `setcodepri()` or `getcodepri()`.

System	Freq.	Signal Priority (1: highest > 10: lowest) *									
		1	2	3	4	5	6	7	8	9	10
GPS	L1	<b>1C</b>	<b>1P</b>	<b>1Y</b>	<b>1W</b>	<b>1M</b>	<b>1N</b>	<b>1S</b>	<b>1L</b>		
	L2	<b>2C</b>	<b>2P</b>	<b>2Y</b>	<b>2W</b>	<b>2M</b>	<b>2N</b>	<b>2D</b>	<b>2L</b>	<b>2S</b>	<b>2X</b>
	L5	<b>5I</b>	<b>5Q</b>	<b>5X</b>							
GLONASS	G1	<b>1C</b>	<b>1P</b>	<b>1A</b>	<b>1B</b>	<b>1X</b>					
	G2	<b>2C</b>	<b>2P</b>	<b>2A</b>	<b>2B</b>	<b>2X</b>					
	G3	<b>3I</b>	<b>3Q</b>	<b>3X</b>							
Galileo	E1	<b>1C</b>	<b>1A</b>	<b>1B</b>	<b>1X</b>	<b>1Z</b>					
	E5a	<b>5I</b>	<b>5Q</b>	<b>5X</b>							
	E5b	<b>7I</b>	<b>7Q</b>	<b>7X</b>							
	E5a+E5b	<b>8I</b>	<b>8Q</b>	<b>8X</b>							
	E6	<b>6A</b>	<b>6B</b>	<b>6C</b>	<b>6X</b>	<b>6Z</b>					
QZSS	L1	<b>1C</b>	<b>1L</b>	<b>1S</b>	<b>1X</b>	<b>1Z</b>					
	L2	<b>2L</b>	<b>2S</b>	<b>2X</b>							
	L5	<b>5I</b>	<b>5Q</b>	<b>5X</b>	<b>5D</b>	<b>5P</b>	<b>5Z</b>				
	LEX	<b>6L</b>	<b>6S</b>	<b>6X</b>	<b>6E</b>	<b>6Z</b>					
BeiDou	B1	<b>2I</b>	<b>2Q</b>	<b>2X</b>	<b>2D</b>	<b>2P</b>	<b>2A</b>	<b>2N</b>			
	B2	<b>7I</b>	<b>7Q</b>	<b>7X</b>	<b>7D</b>	<b>7P</b>	<b>7Z</b>				
	B3	<b>6I</b>	<b>6Q</b>	<b>6X</b>							
SBAS	L1	<b>1C</b>									
	L5	<b>5I</b>	<b>5Q</b>	<b>5X</b>							

\* Refer Appendix D.3 for signal IDs,

## D.5 Receiver Dependent Input Options

Format	Option	Description
RTCM 2	-	-
	<b>-EPHALL</b>	<b>Input all of ephemerides</b>
	<b>-INVPRR</b>	<b>Invert polarity of PhaseRangeRate</b>
	<b>-STA=nnn</b>	<b>Input only messages with STAID=nnn</b>
RTCM 3	<b>-GLss</b>	Select signal ss for GPS MSM (ss=1C,1P...) * **
	<b>-RLss</b>	Select signal ss for GLO MSM (ss=1C,1P...) * **
	<b>-ELss</b>	Select signal ss for GAL MSM (ss=1C,1B...) * **
	<b>-JLss</b>	Select signal ss for QZS MSM (ss=1C,2C...) * **
	<b>-CLss</b>	Select signal ss for BDS MSM (ss=2I,7I...) * **
	<b>-EPHALL</b>	<b>Input all of ephemerides</b>
	<b>-GLss</b>	Select signal ss for GPS (ss=1C,1P...) * **
BINEX	<b>-RLss</b>	Select signal ss for GLO (ss=1C,1P...) * **
	<b>-ELss</b>	Select signal ss for GAL (ss=1C,1B...) * **
	<b>-JLss</b>	Select signal ss for QZS (ss=1C,2C...) * **
	<b>-CLss</b>	Select signal ss for BDS (ss=2I,7I...) * **
	<b>-EPHALL</b>	<b>Input all of ephemerides</b>
NovAtel OEM4/V/6	<b>-GL1P</b>	Select 1P for GPS L1 (default 1C) *
	<b>-GL2X</b>	Select 2X for GPS L2 (default 2W) *
	<b>-RL2C</b>	Select 2C for GLO L2 (default 2P) *
	<b>-EL2C</b>	Select 2C for GAL L2 (default 2C) *
NovAtel OEM3	-	-
	<b>-EPHALL</b>	<b>Input all of ephemerides</b>
	<b>-INVCP</b>	<b>Invert polarity of carrier-phase</b>
	<b>-TADJ=tint</b>	Adjust time tags to multiples of tint (sec)
u-blox 4T/5T/6T/8T 8P/9P	<b>-MULTICODE</b>	Preserve multiple signal codes for single freq
	<b>-RCVSTD5</b>	Save receiver stdevs to unused rinex fields
	<b>-MAX_STD_CP=n</b>	Reject observations with StDev > n - Default for Gen8 (M8T, M8P) n=5 - Default for Gen9 (F9P) n=8
	<b>-STD_SLIP=n</b>	Set cycle slip for phase obs with StDev > n
NovAtel Superstar II	-	-
Hemisphere Crescent/Eclipse	<b>-EPHALL</b>	<b>Input all of ephemerides</b>
SkyTraq S1315F	<b>-INVCP</b>	<b>Invert polarity of carrier-phase</b>
Furuno GW-10-II/III	<b>-EPHALL</b>	<b>Input all of ephemerides</b>
	<b>-EPHALL</b>	<b>Input all of ephemerides</b>
JAVAD GRIL/GRIES	<b>-GL1W</b>	Select 1W for GPS L1 (default 1C) *
	<b>-GL1X</b>	Select 1X for GPS L1 (default 1C) *
	<b>-GL2X</b>	Select 2X for GPS L2 (default 2W) *
	<b>-RL1C</b>	Select 1C for GLO L2 (default 1P) *

Format	Option	Description
	<b>-RL2C</b>	Select 2C for GLO L2 (default 2P) *
	<b>-JL1Z</b>	Select 1Z for QZS L1 (default 1C) *
	<b>-JL1X</b>	Select 1X for QZS L1 (default 1C) *
	<b>-NOET</b>	Discard epoch time message ET (::)
NVS NV08C	<b>-EPHALL</b>	Input all of ephemerides
BINR	<b>-TADJ=tint</b>	Adjust time tags to multiples of tint (sec)

\* Refer Appendix D.3 for signal IDs, \*\* Refer Appendix D.4 for default signal priority

## Appendix E Models and Algorithms

This appendix briefly describes the models and algorithms involved in RTKLIB. The common acronyms used for models or equations in the appendix are shown as follows.

$c$  : speed of light (m/s)

$P_{r,i}^s$  :  $L_i$  pseudorange measurement of signal (m)

$D_{r,i}^s$  :  $L_i$  Doppler frequency measurement of signal (Hz)

$\phi_{r,i}^s$  :  $L_i$  carrier-phase measurement of signal (cycle)

$\Phi_{r,i}^s$  :  $L_i$  phase-range measurement of signal (m)

$t_r$  : navigation signal reception time at receiver (s)

$t^s$  : navigation signal transmission time at satellite (s)

$\rho_r^s$  : geometric range between satellite and receiver antennas (m)

$r_r^s$  : range-rate between satellite and receiver antennas (m/s)

$\mathbf{r}^s(t)$  : satellite position at time  $t$  in ECEF (m)

$\mathbf{v}^s(t)$  : satellite velocity at time  $t$  in ECEF (m)

$\mathbf{r}_r(t)$  : receiver antenna position at time  $t$  in ECEF (m)

$\mathbf{v}_r(t)$  : receiver antenna velocity at time  $t$  in ECEF (m/s)

$\mathbf{e}_r^s$  : LOS vector from receiver antenna to satellite in ECEF

$\mathbf{e}_{r,enu}^s$  : LOS vector from receiver antenna to satellite in local coordinates

$\mathbf{E}_r$  : coordinates rotation matrix from ECEF to local coordinates at receiver antenna position

$\mathbf{E}^s$  : coordinates rotation matrix from ECEF to satellite body-fixed coordinates

$\varphi_r$  : latitude of receiver antenna position (rad)

$\lambda_r$  : longitude of receiver antenna position (rad)

$Az_r^s$  : azimuth angle of satellite direction (rad)

$El_r^s$  : elevation angle of satellite direction (rad)

$dt_r(t)$  : receiver clock bias at time  $t$  (s)

$d\dot{t}_r(t)$  : receiver clock drift at time  $t$  (s/s)

$dT^s(t)$  : satellite clock bias at time  $t$  (s)

$d\dot{T}^s(t)$  : satellite clock drift at time  $t$  (s/s)

$T_r^s$  : tropospheric delay (m)

$I_{r,i}^s$  :  $L_i$  ionospheric delay (m)

$f_i$  :  $L_i$  carrier frequency (Hz)

$\lambda_i$  :  $L_i$  carrier wave length (m)

$B_{r,i}^s$  :  $L_i$  carrier-phase bias (cycle)

- $N_{r,i}^s : L_i$  carrier-phase integer ambiguity (cycle)
- $\varepsilon_P : \text{measurement errors of pseudorange (m)}$
- $\varepsilon_\phi : \text{measurement errors of carrier-phase (cycle)}$
- $\varepsilon_\Phi : \text{measurement errors of phase-range (m)}$
- $\omega_e : \text{earth rotation angle velocity (rad/s)}$
- $\mathbf{d}_{r,pco,i} : L_i$  receiver antenna phase center offset in local coordinates (m)
- $\mathbf{d}_{pco,i}^s : L_i$  satellite antenna phase center offset in satellite body-fixed coordinates (m)
- $d_{r,pcv,i}(El) : L_i$  receiver antenna phase center variation (m)
- $d_{pcv,i}^s(\theta) : L_i$  satellite antenna phase center variation (m)
- $\alpha : \text{off-nadir angle of receiver antenna direction from satellite (rad)}$
- $\mathbf{d}_{r,disp} : \text{displacement by earth tides at receiver position in local coordinates (m)}$
- $\phi_{pw} : \text{phase windup effect (cycle)}$
- $dT_{rel}^s : \text{relativity correction for satellite clock (s)}$
- $\mathbf{U}(t) : \text{ECI to ECEF coordinates transformation matrix at time } t$
- $\mathbf{R}_x(\theta) : \text{coordinates rotation matrix around x-axis by angle } \theta$
- $\mathbf{R}_y(\theta) : \text{coordinates rotation matrix around y-axis by angle } \theta$
- $\mathbf{R}_z(\theta) : \text{coordinates rotation matrix around z-axis by angle } \theta$
- $\bigtriangleup^{jk} : \text{SD (single-difference) between satellite } j \text{ and } k$
- $\bigtriangleup : \text{SD (single-difference) between receiver } r \text{ and } b$
- $Z_{T,r} : \text{tropospheric zenith total delay (m)}$
- $Z_{H,r} : \text{tropospheric zenith hydrostatic delay (m)}$
- $G_{E,r} : \text{east component of tropospheric gradient}$
- $G_{N,r} : \text{north component of tropospheric gradient}$
- $m_{H,r}^s : \text{mapping function for hydrostatic tropospheric delay}$
- $m_{W,r}^s : \text{mapping function for wet tropospheric delay}$
- $m_{I,r}^s : \text{mapping function for ionospheric delay}$

## E.1 Time System

RTKLIB internally uses GPST (GPS Time) for GNSS data handling and positioning algorithms. The time of input data expressed in other time systems like UTC (Universal Time Coordinated) is converted to GPST before internal processing or the GPST of the internal data is converted to the appropriate other time system before output. One of the reasons why using GPST is to avoid leap seconds handling. UTC, which is the most generally used time system, is not a continuous time system with a second jump by the leap second insertion or deletion.

The GPST is often expressed as a GPS week number and TOW (time of week) in seconds since the start epoch of 00:00:00 UTC on January 6, 1980. RTKLIB, however, does not use such a convention. In GNSS data processing, we often need to convert a time to a range or a range to a time. The TOW even expressed as a double precision has only the resolution of  $1.3 \times 10^{-10}$  s in time, which is equivalent to the resolution of 0.04 m in range. So, instead, RTKLIB expresses time as the type **gtime\_t** defined as:

```
typedef struct {
    time_t time; /* time (s) expressed by standard time_t */
    double sec; /* fraction of second under 1 s */
} gtime_t;
```

where the **time\_t** is the standard time type provided by the standard C library. The internal representation of the **time\_t** depends upon the computer system. It is often implemented by an unsigned 32 bit integer as the number of total seconds since 00:00:00 on January 1, 1970. Due to the bit length limitation the integer, the **gtime\_t** cannot handle the time before January 1, 1970 or after January 19, 2038 in this case.

The **sec** of the **gtime\_t** holds the fractional second. The **sec** has the resolution of  $6.7 \times 10^{-8}$  m in range adequate for GNSS precise position computation. RTKLIB also provides several useful APIs to handle the **gtime\_t** including adding, difference, conversion to/from the GPS week number and TOW.

### (1) GPST and UTC (Universal Time Coordinated)

The rough conversion of GPST to UTC (Universal Time Coordinated) or UTC to GPST can be expressed simply as:

$$t_{UTC} = t_{GPST} - \Delta t_{LS} \quad (E.1.1)$$

$$t_{GPST} = t_{UTC} + \Delta t_{LS} \quad (E.1.2)$$

where  $t_{UTC}$  and  $t_{GPST}$  are the time expressed in UTC (s) and the time in GPST (s).  $\Delta t_{LS}$  is the delta time (s) between UTC and GPST due to the cumulative leap seconds since January 6, 1980. The values of  $\Delta t_{LS}$  are shown in Table E.1-1 until March 2013.

Table E.1-1 GPST-UTC Values (until March 2013)

Time Since (in UTC)	$\Delta t_{LS}$ (s)
1980-01-06 00:00:00	0
1981-07-01 00:00:00	1
1982-07-01 00:00:00	2
1983-07-01 00:00:00	3
1987-01-01 00:00:00	4
1988-01-01 00:00:00	5
1990-01-01 00:00:00	6
1991-01-01 00:00:00	7
1992-07-01 00:00:00	8
1993-07-01 00:00:00	9
1994-07-01 00:00:00	10
1996-01-01 00:00:00	11
1999-01-01 00:00:00	13
1997-07-01 00:00:00	12
2006-01-01 00:00:00	14
2009-01-01 00:00:00	15
2012-07-01 00:00:00	16

The accuracy of the approximation in (E.1.1) or (E.1.2) is within several 10 ns. By using the UTC parameters in GPS navigation messages, we can convert GPST to UTC or UTC to GPST more accurately as:

$$t_{UTC} = t_{GPS} - \{\Delta t_{LS} + A_0 + A_1(t_E - t_{ot} + 604800(WN - WN_t))\} \quad (\text{E.1.3})$$

where  $A_0$ ,  $A_1$ ,  $t_E$ ,  $t_{ot}$ ,  $WN$  are  $WN_t$  are the UTC parameters provided in GPS navigation messages. More strictly, UTC in (E.1.3) is UTC(USNO), which is the US local implementation of UTC. The difference between UTC and UTC(USNO) can be obtained in Circular T provided by BIPM [72]. The difference is usually several ns level.

## (2) GLONASST (GLONASS Time)

GLONASST (GLONASS Time) is based on UTC(SU) and includes leap second insertion or deletion. GLONASST is also aligned to the local time. So, roughly, the time  $t_{GLONASST}$  (s) in GLONASST can be converted to the time  $t_{UTC}$  (s) in UTC.

$$t_{UTC} = t_{GLONASST} - 10800 \quad (\text{E.1.4})$$

More accurately, the UTC parameters for GLONASST in GLONASS navigation message should be used similar to the GPST and UTC conversion. Ignoring the leap seconds and the 3 hour offset, the difference between GPST and GLONASST is usually 100 or several 100 ns level.

(3) GST (Galileo System Time)

GST (Galileo System Time) is composed of week number from the origin of the Galileo time and the TOW (time of week) in seconds. The GST start epoch is 00:00:00 UTC on August 22, 1999. At the start epoch, GST shall be ahead of UTC by 13 seconds. The GST is continuous time without leap second insertion or deletion. So, the GST is aligned to GPST except for the 1024 weeks difference of the time system origin and a small time offset (GGTO). Note that the Galileo week number is provided as equal to the GPS week number in the RINEX convention.

(4) QZSST (QZSS Time)

QZSST (QZSS Time) is aligned to GPST. It has the same origin as GPST and the same definition of one-second of GPST. Practically, QZSST can be handled as same as GPST.

(5) BDT (BeiDou Navigation Satellite System Time)

BDT (BeiDou Navigation Satellite System Time) is a continuous time system without leap second insertion or deletion. The start epoch of BDT is 00:00:00 UTC on January 1, 2006. The offset of BDT with respect to UTC is controlled within 100 ns (modulo 1 second). So, the time  $t_{GPST}$  (s) in GPST can roughly be converted to the time  $t_{BDT}$  (s) in BDT within the accuracy of 200 ns as:

$$t_{BDT} = t_{GPST} - 14 \quad (\text{E.1.5})$$

More accurately, the UTC parameters for BDT in BeiDou navigation messages should be used similar to the GPST and UTC conversion.

## E.2 Coordinates System

The receiver or satellite positions in RTKLIB are internally represented as the X, Y, Z components in an ECEF (earth center earth fixed) coordinates system. What ECEF frame used is not explicitly defined but depends on the satellite ephemeris and the predefined base station position. For example, with GPS signals and navigation data, the single point positioning results are obtained in WGS84. The baseline analysis with the base station with the position in an ECEF frame basically brings the rover position in the same ECEF frame. Practically, all of usually used ECEF frames in GNSS navigation processing like WGS 84, PZ90.02 and ITRF, are identical within the accuracy of broadcast ephemeris or single point positioning. However, more strict and careful handling of the coordinates system is needed for the baseline analysis or PPP. The unified coordinates system is desirable to minimize the processing error in these cases.

(1) Transformation from geodetic position to ECEF XYZ position

The geodetic position are defined based on a reference ellipsoid shown in Figure E.2-1. The geodetic latitude  $\phi_r$ , longitude  $\lambda_r$  and the ellipsoidal height  $h$  can be transformed to the ECEF XYZ position  $\mathbf{r}_r = (x, y, z)^T$  as follows:

$$e^2 = f(2 - f) \quad (E.2.1)$$

$$v = \frac{a}{\sqrt{1-e^2 \sin \phi_r^2}} \quad (E.2.2)$$

$$\mathbf{r}_r = \begin{pmatrix} (v + h) \cos \phi_r \cos \lambda_r \\ (v + h) \cos \phi_r \sin \lambda_r \\ v(1 - e^2) \sin \phi_r \end{pmatrix} \quad (E.2.3)$$

where:

$a$  : major radius of the earth reference ellipsoid (m)

$f$  : flattening of the earth reference ellipsoid

Current version RTKLIB always uses the following values for  $a$  and  $f$  of the reference ellipsoid provided by the WGS84 datum.

$$a = 6378137.0 \text{ (m)}$$

$$f = 1.0/298.257223563$$

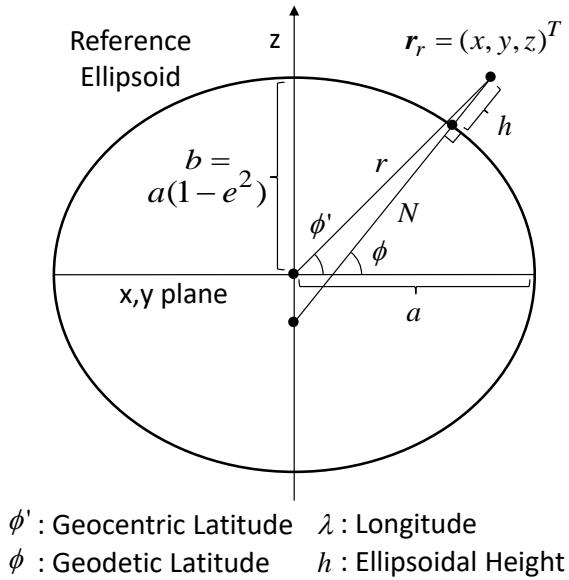


Figure E.2-1 Reference Ellipsoid

## (2) Transformation from ECEF XYZ position to geodetic position

To transform the XYZ position  $\mathbf{r}_r = (x, y, z)^T$  in ECEF to the geodetic position, the following procedure is applied. The geodetic latitude is derived by an iterative method in the procedure.

$$r = \sqrt{x^2 + y^2} \quad (\text{E.2.4})$$

$$\phi_{r,0} = 0 \quad (\text{E.2.5})$$

$$\phi_{r,i+1} = \arctan\left(\frac{z}{r} - \frac{ae^2 \tan \phi_{r,i}}{r\sqrt{1+(1-e^2)\tan^2 \phi_{r,i}}}\right) \quad (\text{E.2.6})$$

$$\phi_r = \lim_{i \rightarrow \infty} \phi_{r,i} \quad (\text{E.2.7})$$

$$\lambda = \text{ATAN2}(y, x) \quad (\text{E.2.8})$$

$$h = \frac{r}{\cos \phi_r} - \frac{a}{\sqrt{(1-e^2)\sin^2 \phi_r}} \quad (\text{E.2.9})$$

## (3) Transformation between local coordinates and ECEF

The local coordinates at the receiver position, which is also called as ENU coordinates, is often used in GNSS navigation processing. The rotation matrix  $\mathbf{E}_r$  of the ECEF coordinates to the local coordinates are expressed as:

$$\mathbf{E}_r = \begin{pmatrix} -\sin \lambda_r & \cos \lambda_r & 0 \\ -\sin \phi_r \cos \lambda_r & -\sin \phi_r \sin \lambda_r & \cos \phi_r \\ \cos \phi_r \cos \lambda_r & \cos \phi_r \sin \lambda_r & \sin \phi_r \end{pmatrix} \quad (\text{E.2.10})$$

where  $\phi_r$  and  $\lambda_r$  are the geodetic latitude and the longitude of the receiver position. By using the  $\mathbf{E}_r$  and the receiver position  $\mathbf{r}_r$  in the ECEF coordinates, the position  $\mathbf{r}_{ECEF}$  in the ECEF coordinates can be transformed to the position  $\mathbf{r}_{local}$  in the local coordinates as:

$$\mathbf{r}_{local} = \mathbf{E}_r(\mathbf{r}_{ECEF} - \mathbf{r}_r) \quad (\text{E.2.11})$$

#### (4) Geoid models

To obtain the geodetic height derived from the ellipsoidal height, we should consider the geoid height.

The geodetic height  $h_{geod}$  (m) is expressed as:

$$h_{geod} = h - geod(\phi_r, \lambda_r) \quad (\text{E.2.12})$$

where  $geod(\phi_r, \lambda_r)$  is the geoid height (m) at the latitude  $\phi_r$  and the longitude  $\lambda_r$ . RTKLIB supports the following geoid models selectable by the processing option "Geoid Model". All of these geoid models are provided as the geoid heights at the latitude and longitude grid points. RTKLIB uses bilinear interpolation to obtain the value for an appropriate latitude and longitude position. To use these geoid model except for the internal model, the geoid model data file should be downloaded and the file path should be set as the processing option "Geoid Data File".

- (a) Internal : 1 deg x 1 deg grid geoid derived from EGM96
- (b) EGM96 : 15" x 15" grid EGM96 geoid model
- (c) EGM2008 : 2.5" x 2.5" grid or 1" x 1" grid EGM2008 geoid model
- (d) GSI 2000 : 1" x 1.5" grid GSI 2000 geoid model (only surrounding of Japanese island)

### E.3 GNSS Signal Measurement Models

#### (1) GNSS Signal Structure

Figure E.3-1 shows a typical GNSS signal structure. The GNSS signal is generally composed of multiplication of the carrier frequency (Carrier), the spreading code (Code) and the navigation data (Data). The spreading codes are also called as PRN (pseudo random noise) codes. The detailed specifications of these GNSS signals provided by GPS, GLONASS, Galileo, QZSS, BeiDou and SBAS are found in Appendix G.

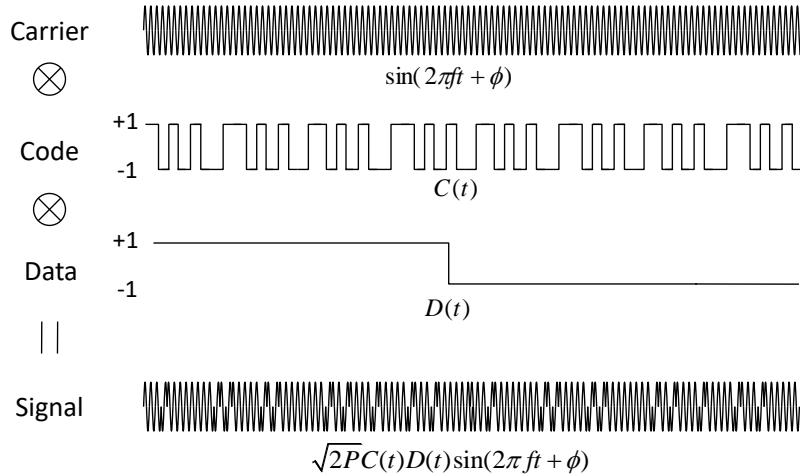


Figure E.3-1 GNSS Signal Structure

#### (2) Pseudorange measurement model

The pseudorange is defined as "the distance from the receiver antenna to the satellite antenna including receiver and satellite clock offsets (and other biases, such as atmospheric delays)" [9]. The  $L_i$  pseudorange  $P_{r,i}^s$  can be expressed by using the signal reception time  $\bar{t}_r$  (s) measured by the receiver clock and the signal transmission time  $\bar{t}^s$  (s) measured by the satellite clock as:

$$P_{r,i}^s = c(\bar{t}_r - \bar{t}^s) \quad (\text{E.3.1})$$

The equation can be written by using the geometric range  $\rho_r^s$  between satellite and receiver antennas, the receiver and satellite clock biases  $dt_r, dT^s$ , the ionospheric and tropospheric delay  $I_{r,i}^s, T_r^s$  and the measurement error  $\varepsilon_P$  as: [64]

$$\begin{aligned}
P_{r,i}^s &= c((t_r + dt_r(t_r)) - (t^s + dT^s(t^s))) + \varepsilon_p \\
&= c(t_r - t^s) + c(dt_r(t_r) - dT^s(t^s)) + \varepsilon_p \\
&= (\rho_r^s + I_{r,i}^s + T_r^s) + c(dt_r(t_r) - dT^s(t^s)) + \varepsilon_p \\
&= \rho_r^s + c(dt_r(t_r) - dT^s(t^s)) + I_{r,i}^s + T_r^s + \varepsilon_p
\end{aligned} \tag{E.3.2}$$

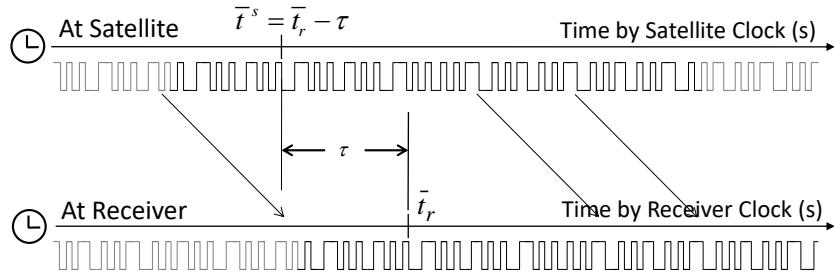


Figure E.3-2 Pseudorange Model

### (3) Carrier-phase and phase-range measurement model

The carrier-phase is "... actually being a measurement on the beat frequency between the received carrier of the satellite signal and a receiver-generated reference frequency" [9]. The  $L_i$  carrier-phase  $\phi_{r,i}^s$  can be expressed as:

$$\begin{aligned}
\phi_{r,i}^s &= \phi_{r,i}(t_r) - \phi_i^s(t^s) + N_{r,i}^s + \varepsilon_\phi \\
&= (f_i(t_r + dt_r(t_r) - t_0) + \phi_{r,0,i}) - (f_i(t^s + dT^s(t^s) - t_0) + \phi_{0,i}^s) + N_{r,i}^s + \varepsilon_\phi \\
&= \frac{c}{\lambda_i} (t_r - t^s) + \frac{c}{\lambda_i} (dt_r(t_r) - dT^s(t^s)) + (\phi_{r,0,i} - \phi_{0,i}^s + N_{r,i}^s) + \varepsilon_\phi
\end{aligned} \tag{E.3.3}$$

where  $t_0$  is the initial time (s),  $\phi_{r,i}(t)$  is the  $L_i$  phase (cycle) of receiver local oscillator and  $\phi_i^s(t)$  is the  $L_i$  phase (cycle) of transmitted navigation signal at the time  $t$ .  $\phi_{r,0,i}$  is the  $L_i$  initial phase (cycle) of receiver local oscillator and  $\phi_{0,i}^s$  is the  $L_i$  initial phase (cycle) of transmitted navigation signal at the time  $t_0$ .

The  $L_i$  phase-range  $\Phi_{r,i}^s$ , defined as the carrier-phase multiplied by the carrier frequency  $\lambda_i$  in m, also can be expressed by using the carrier phase bias  $B_{r,i}^s$  and carrier-phase correction terms  $d\phi_{r,i}^s$  including

antenna phase center offsets and variations, station displacement by earth tides, phase windup effect and relativity correction on the satellite clock as:

$$\begin{aligned}\Phi_{r,i}^s &= \lambda_i \phi_{r,i}^s \\ &= c(t_r - t^s) + c(dt_r(t_r) - dT^s(t^s)) + \lambda_i(\phi_{r,0,i} - \phi_{0,i}^s + N_{r,i}^s) + \lambda_i \varepsilon_\phi \\ &= \rho_r^s + c(dt_r(t_r) - dT^s(t^s)) - I_{r,i}^s + T_r^s + \lambda_i B_{r,i}^s + d\Phi_{r,i}^s + \varepsilon_\phi\end{aligned}\quad (\text{E.3.4})$$

where:

$$B_{r,i}^s = \phi_{r,0,i} - \phi_{0,i}^s + N_{r,i}^s \quad (\text{E.3.5})$$

$$d\Phi_{r,i}^s = -\mathbf{d}_{r,pco,i}^T \mathbf{e}_{r,enu}^s + (\mathbf{E}^s \mathbf{d}_{pco,i}^s)^T \mathbf{e}_r^s + d_{r,pcv,i}(El) + d_{pcv,i}^s(\theta) - \mathbf{d}_{r,disp}^T \mathbf{e}_{r,enu}^s + \lambda_i \phi_{pw} \quad (\text{E.3.6})$$

$N_{r,i}^s$  is often called as carrier-phase integer ambiguity, carrier-cycle ambiguity or simply ambiguity.

For the detailed formulation of the carrier-phase correction terms, refer Appendix E.9.

#### (4) Geometric range between receiver and satellite antennas

The geometric range is defined as the physical distance between the satellite antenna phase center position and the receiver antenna phase center position in the inertial coordinates. At first, the signal transmission time  $t^s$  can be derived from:

$$t^s = \bar{t}_r - P_{r,i}^s/c - dT(t^s) \quad (\text{E.3.7})$$

The both sides in the equation includes  $t^s$ . So several iterations are needed to solve the equation. The geometric range can be expressed by using the receiver and satellite antenna phase center positions  $\mathbf{r}_r(t_r) = (x_r, y_r, z_r)^T$  at the time  $t_r$  and  $\mathbf{r}^s(t^s) = (x^s, y^s, z^s)^T$  at the time  $t^s$  in the ECEF (earth center earth fixed) coordinates as:

$$\rho_r^s = \|\mathbf{U}(t_r)\mathbf{r}_r(t_r) - \mathbf{U}(t^s)\mathbf{r}^s(t^s)\| \quad (\text{E.3.8})$$

where  $\mathbf{U}(t)$  is the ECEF to ECI (earth center inertial) coordinates transformation matrix at the time  $t$ . For the expression in the ECEF coordinates, the earth rotation effect has to be incorporated in to obtain the geometric range. The equation can be approximated by one of the following equations with adequate precision under 1 mm level. Current version RTKLIB always uses the equation (F.3.8b) for the geometric range. The last term in (F.3.8b) is sometimes called as Sagnac effect.

$$\rho_r^s \approx \|\mathbf{r}_r(t_r) - \mathbf{R}_z(\omega_e \rho_r^s/c) \mathbf{r}^s(t^s)\| \quad (\text{E.3.8a})$$

$$\rho_r^s \approx \|\mathbf{r}_r(t_r) - \mathbf{r}^s(t^s)\| + \frac{\omega_e}{c} (x^s y_r - y^s x_r) \quad (\text{E.3.8b})$$

$$\rho_r^s \approx \|\mathbf{r}_r(t_r) - \mathbf{R}_z(\omega_e(t_r - t^s)) \mathbf{r}^s(t^s)\| \quad (\text{E.3.8c})$$

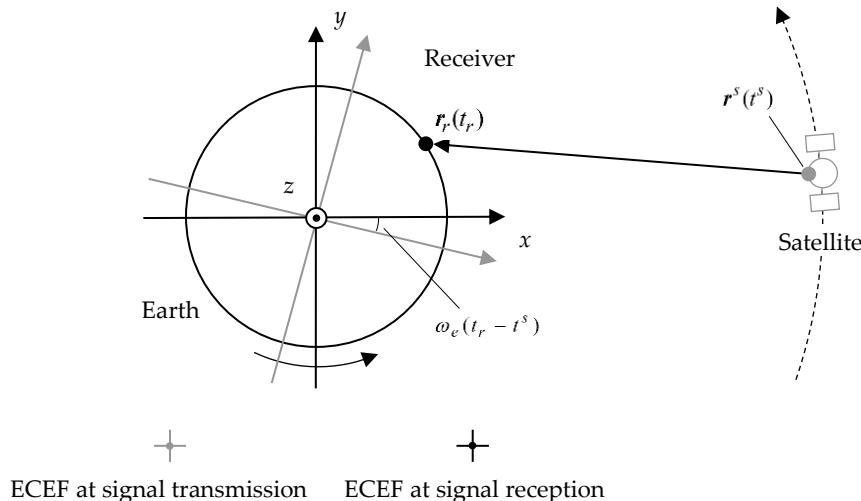


Figure E.3-1 Geometric Range and Earth Rotation Correction

## (5) Azimuth and elevation angles of satellite direction

The unit LOS (line-of-sight) vector from the receiver to the satellite can be expressed in the ECEF coordinates as:

$$\mathbf{e}_r^s = \frac{\mathbf{r}^s(t^s) - \mathbf{r}_r(t_r)}{\|\mathbf{r}^s(t^s) - \mathbf{r}_r(t_r)\|} \quad (\text{E.3.9})$$

In the equation, the earth rotation effect is neglected. The azimuth and elevation angles  $Az_r^s$  and  $El_r^s$  of the satellite direction from the receiver site can be derived from:

$$\mathbf{e}_{r,enu}^s = \mathbf{E}_r \mathbf{e}_r^s = (e_e, e_n, e_u)^T \quad (\text{E.3.10})$$

$$Az_r^s = ATAN2(e_e, e_n) \quad (\text{E.3.11})$$

$$El_r^s = arcsin(e_u) \quad (\text{E.3.12})$$

where  $\mathbf{E}_r$  is the coordinates rotation matrix from ECEF to the local coordinates at the receiver position. Refer E.2 for detailed formation of the matrix.

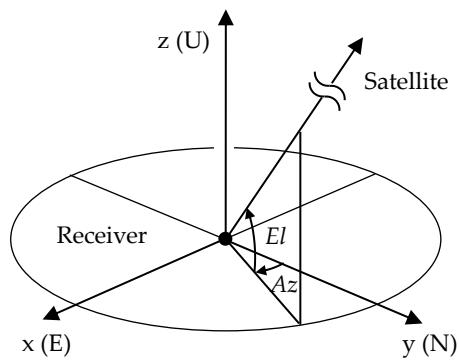


Figure E.3-2 Local Coordinates and Azimuth and Elevation Angles

## E.4 GNSS Satellite Ephemerides and Clocks

RTKLIB supports broadcast ephemerides and clocks for GPS, GLONASS, Galileo, QZSS, BeiDou and SBAS. It also supports the precise ephemerides and clocks provided as the SP3-c [22] and clock RINEX [15] including Galileo, QZSS and BeiDou for post processing modes. For real-time modes, the broadcast ephemerides and clocks corrected by the SBAS long-term and fast corrections and the RTCM 3 SSR (state space representation) corrections are also supported. The following equations show the ephemeris and clock models used in RTKLIB.

- (1) Broadcast ephemerides and clocks for GPS, Galileo and QZSS [1] [5] [6]

Broadcast ephemeris and SV clock parameters for GPS, Galileo and QZSS are given in navigation messages as:

$$\mathbf{p}_{eph}(t_{oe}, t_{oc}, IOD) = (a, e, i_0, \Omega_0, \omega, M_0, \Delta n, \dot{I}, \dot{\Omega}, C_{us}, C_{uc}, C_{rs}, C_{rc}, C_{is}, C_{ic}, af_0, af_1, af_2, T_{GD})^T \quad (\text{E.4.1})$$

By using these parameters, the satellite position (antenna phase center position)  $\mathbf{r}^s(t)$  in ECEF, the satellite clock bias  $dT^s(t)$  and clock drift  $d\dot{T}^s(t)$  are computed as:

$$t_k = t - t_{oe} \quad (\text{E.4.2})$$

$$M = M_0 + \left( \sqrt{\frac{\mu}{a^3}} + \Delta n \right) t_k \quad (\text{E.4.3})$$

$$M = E - e \sin E \quad (\text{E.4.4})$$

$$\alpha = \frac{\sqrt{1-e^2} \sin E}{\cos E - e} \quad (\text{E.4.5})$$

$$\phi = \arctan \alpha + \omega \quad (\text{E.4.6})$$

$$\delta u = C_{us} \sin 2\phi + C_{uc} \cos 2\phi \quad (\text{E.4.7})$$

$$\delta r = C_{rs} \sin 2\phi + C_{rc} \cos 2\phi \quad (\text{E.4.8})$$

$$\delta i = C_{is} \sin 2\phi + C_{ic} \cos 2\phi \quad (\text{E.4.9})$$

$$u = \phi + \delta u \quad (\text{E.4.10})$$

$$r = a(1 - e \cos E) + \delta r \quad (\text{E.4.11})$$

$$i = i_0 + \delta i + \dot{I}t_k \quad (\text{E.4.12})$$

$$\Omega = \Omega_0 + (\dot{\Omega} - \omega_e)t_k - \omega_e t_{oe} \quad (\text{E.4.13})$$

$$\mathbf{r}^s(t) = r \begin{pmatrix} \cos u \cos \Omega - \sin u \cos i \sin \Omega \\ \cos u \sin \Omega + \sin u \cos i \cos \Omega \\ \sin u \sin i \end{pmatrix} \quad (\text{E.4.14})$$

$$t_c = t - t_{oc} \quad (\text{E.4.15})$$

$$dT^s(t) = af_0 + af_1 t_c + af_2 t_c^2 - \frac{2\sqrt{\mu}}{c^2} e \sqrt{A} \sin E - b T_{GD} \quad (\text{E.4.16})$$

$$d\dot{T}^s(t) = af_1 + 2af_2 t_c \quad (\text{E.4.17})$$

where:

$\mu$  : earth gravitational constant ( $3.9860050 \times 10^{14} \text{m}^3/\text{s}^2$  for GPS and QZSS,  $3.986004418 \times 10^{14} \text{ m}^3/\text{s}^2$  for Galileo)

$\omega_e$  : earth angular velocity ( $7.2921151467 \times 10^{-5}$ rad/s)

$b = f_1^2/f_i^2$  for  $L_i$  pseudorange

$T_{GD}$  : group delay parameters for GPS and QZSS,  $B_{GD}$  for Galileo (s)

The Kepler equation (E.4.4) can be solved the following iteration by Newton's method.

$$E_0 = M \quad (\text{E.4.18})$$

$$E_{i+1} = E_i - \frac{E_i - e \sin E_i - M}{1 - e \cos E_i} \quad (\text{E.4.19})$$

$$E = \lim_{i \rightarrow \infty} E_i \quad (\text{E.4.20})$$

The broadcast ephemerides and clock are applied in case that the processing option "Satellite Ephemeris/Clock" to "Broadcast" as well as GLONASS, BeiDou and SBAS.

## (2) Broadcast ephemerides and clocks for GLONASS [4]

Broadcast ephemeris and clock parameters for GLONASS are given in the navigation messages as:

$$\mathbf{p}_{eph}(t_b) = (x, y, z, v_x, v_y, v_z, a_x, a_y, a_z, \tau_n, \gamma_n) \quad (\text{E.4.21})$$

The differential equations for the satellite position  $\mathbf{r}^s(t) = (x, y, z)^T$  and velocity  $\mathbf{v}^s(t) = (v_x, v_y, v_z)^T$  in ECEF (PZ90.02) can be formed as:

$$\frac{dx}{dt} = v_x, \frac{dy}{dt} = v_y, \frac{dz}{dt} = v_z \quad (\text{E.4.22})$$

$$\frac{dv_x}{dt} = -\frac{\mu}{r^3}x - \frac{3}{2}J_2 \frac{\mu a_e^2}{r^5}x \left(1 - \frac{5z^2}{r^2}\right) + \omega_e^2 x + 2\omega_e v_y + a_x \quad (\text{E.4.23})$$

$$\frac{dv_y}{dt} = -\frac{\mu}{r^3}y - \frac{3}{2}J_2 \frac{\mu a_e^2}{r^5}y \left(1 - \frac{5z^2}{r^2}\right) + \omega_e^2 y - 2\omega_e v_x + a_y \quad (\text{E.4.24})$$

$$\frac{dv_z}{dt} = -\frac{\mu}{r^3}z - \frac{3}{2}J_2 \frac{\mu a_e^2}{r^5}z \left(3 - \frac{5z^2}{r^2}\right) + a_z \quad (\text{E.4.25})$$

where:

$a_e$ : earth semi-major axis (6378136.0m)

$\mu$  : earth gravitational constant ( $398600.44 \times 10^9$ m<sup>3</sup>/s<sup>2</sup>)

$\omega_e$  : earth angular velocity ( $7.292115 \times 10^{-5}$ rad/s)

$J_2$  : second zonal harmonic of the geopotential ( $1082625.7 \times 10^{-9}$ )

$$r = \sqrt{x^2 + y^2 + z^2}$$

Note that two errata in A.3.1.2 of GLONASS ICD 5.1 [4] has been corrected in the models above.

The satellite position  $\mathbf{r}^s(t)$  and velocity  $\mathbf{v}^s(t)$  at the time  $t$  can be derived by the RK4 (Runge-Kutta 4th order and stage) numerical integration to solve these differential equations with the initial satellite position  $\mathbf{r}^s(t_b)$  and velocity  $\mathbf{v}^s(t_b)$  at the reference time  $t_b$ . For satellite clock bias  $dT^s(t)$  and drift  $d\dot{T}^s(t)$  at the epoch time  $t$  are also derived as:

$$dT^s(t) = -\tau_n + \gamma_n(t - t_b) \quad (\text{E.4.26})$$

$$d\dot{T}^s(t) = \gamma_n \quad (\text{E.4.27})$$

The relativistic effect in the satellite clock are included in the GLONASS clock parameters. So the relativistic correction is not applied in this case.

(3) Broadcast ephemerides and clocks for BeiDou [7]

For BeiDou satellites, the similar ephemeris and clock parameters as GPS, Galileo and QZSS are provided in the navigation messages as:

$$\mathbf{p}_{eph}(t_{oe}, t_{oc}) = (a, e, i_0, \Omega_0, \omega, M_0, \Delta n, \dot{I}, \dot{\Omega}, C_{us}, C_{uc}, C_{rs}, C_{rc}, C_{is}, C_{ic}, af_0, af_1, af_2, T_{GD})^T \quad (\text{E.4.28})$$

For MEO and IGSO satellites of BeiDou, the same formulations as (1) for GPS ephemeris and clock, except for  $\mu = 3.986004418 \times 10^{14}$ ,  $\omega_e = 7.2921150 \times 10^{-5}$  rad/s and the time  $t$  is expressed in BDT.

To obtain the satellite position  $\mathbf{r}^s(t)$  of BeiDou GEO satellites at the time  $t$  in BDT, the equation (E.5.13) and (E.5.14) should be replaced by:

$$\Omega = \Omega_0 + \dot{\Omega}t_k - \omega_e t_{oe} \quad (\text{E.4.29})$$

$$\mathbf{r}^s(t) = r \mathbf{R}_z(\omega_e t_k) \mathbf{R}_x(-5^\circ) \begin{pmatrix} \cos u \cos \Omega - \sin u \cos i \sin \Omega \\ \cos u \sin \Omega + \sin u \cos i \cos \Omega \\ \sin u \sin i \end{pmatrix} \quad (\text{E.4.30})$$

where:

$$\mathbf{R}_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{pmatrix}, \mathbf{R}_z(\theta) = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

(4) Broadcast ephemerides and clocks for SBAS [8]

Navigation message parameters for SBAS GEO satellites are given in the SBAS messages (message type 9) as:

$$\mathbf{p}_{eph}(t_0) = (x, y, z, v_x, v_y, v_z, a_x, a_y, a_z, a_{GF0}, a_{GF1}) \quad (\text{E.4.31})$$

By using these parameters, the satellite position  $\mathbf{r}^s(t)$  in ECEF and satellite clock bias  $dT^s(t)$  are computed as:

$$\mathbf{r}^s(t) = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} v_x \\ v_y \\ v_z \end{pmatrix}(t - t_0) + \frac{1}{2} \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix}(t - t_0)^2 \quad (\text{E.4.32})$$

$$dT^s(t) = a_{GF0} + a_{GF1}(t - t_0) \quad (\text{E.4.33})$$

## (5) SBAS orbit and clock corrections [8]

The SBAS orbit and clock corrections are defined as the following parameters.

$$\Delta_{sbas}(t_0, IOD) = (\delta x, \delta y, \delta z, \delta \dot{x}, \delta \dot{y}, \delta \dot{z}, \delta af_0, \delta af_1) \quad (\text{E.4.38})$$

The *IOD* indicates the target broadcast ephemeris and clock parameters. The corrected satellite position  $\mathbf{r}^s(t)$  at time  $t$  is computed as:

$$\mathbf{r}^s(t) = \mathbf{r}^s_{broadcast}(t) + \begin{pmatrix} \delta x \\ \delta y \\ \delta z \end{pmatrix} + \begin{pmatrix} \delta \dot{x} \\ \delta \dot{y} \\ \delta \dot{z} \end{pmatrix}(t - t_0) \quad (\text{E.4.39})$$

where:

$\mathbf{r}^s_{broadcast}(t)$  : satellite position at time  $t$  computed by the broadcast ephemeris with *IOD* (m)

The corrected satellite clock bias  $dT^s(t)$  at the time  $t$  is also computed with the SSR correction as:

$$dT^s(t) = dT^s_{broadcast}(t) + \delta af_0 + \delta af_1(t - t_0) \quad (\text{E.4.40})$$

where:

$dT^s_{broadcast}(t)$  : satellite clock bias at time  $t$  computed by the broadcast clock with *IOD*.

The SBAS correction with broadcast ephemerides and clocks are applied in case that the processing option "Satellite Ephemeris/Clock" to "Broadcast+SBAS".

## (6) Precise ephemerides and clocks

The precise ephemerides for GPS, GLONASS, Galileo, QZSS and BeiDou are usually provided as SP3-c files containing satellite positions and velocities (optional) at every 15 min or 5 min epochs. To obtain the satellite position at the time  $t$ , an appropriate interpolation is needed. RTKLIB uses the fixed degree ( $n = 10$ ) polynomial interpolation by Newton-Neville's algorithm as:

$$P_{j,j}(t) = x_j \quad (i \leq j \leq i + n) \quad (\text{E.4.34})$$

$$p_{j,k}(t) = \frac{(t_k - t)p_{j,k-1}(t) + (t - t_j)p_{j+1,k}(t)}{t_k - t_j} \quad (i \leq j < k \leq i + n) \quad (\text{E.4.35})$$

where  $n$  is the degree of the polynomial for the interpolation and  $x(t_i), x(t_{i+1}), x(t_{i+2}), \dots, x(t_{i+n})$  are

the ephemeris values for each components at the epochs  $t_i, t_{i+1}, t_{i+2}, \dots, t_{i+n}$ . For example, in the  $n = 4$  case, the interpolated value  $x(t)$  at the time  $t$  can be derived as:

$$\begin{aligned} p_{i,i}(t) &= x(t_i) \\ p_{i,i+1}(t) &= x(t_{i+1}) \\ p_{i+1,i+1}(t) &= x(t_{i+1}) p_{i,i+2}(t) \\ p_{i+1,i+2}(t) &= x(t_{i+2}) p_{i+1,i+3}(t) \\ p_{i+2,i+2}(t) &= x(t_{i+2}) p_{i+1,i+3}(t) p_{i,i+3}(t) \\ p_{i+1,i+3}(t) &= x(t_{i+3}) p_{i+2,i+4}(t) \\ p_{i+3,i+3}(t) &= x(t_{i+3}) p_{i+2,i+4}(t) p_{i,i+4}(t) \\ p_{i+4,i+4}(t) &= x(t_{i+4}) \end{aligned}$$

Note that precise ephemerides usually present the CoM (center of mass) positions of satellite not as the antenna phase center position. So users should correct the satellite antenna phase center offset to use the precise ephemerides. For details, refer Appendix E.8.

In spite of the precise ephemeris high-order polynomial interpolation, a simple linear interpolation is implemented for precise clocks provided as SP3-c or clock RINEX files as:

$$dT^s(t) = \frac{(t_{i+1}-t)dT^s(t_i)+(t-t_i)dT^s(t_{i+1})}{t_{i+1}-t_i} \quad (t_i \leq t < t_{i+1}) \quad (\text{E.4.36})$$

For the precise clocks provided by IGS (International GNSS service), the relativistic effect should be corrected as: [68]

$$dT^s(t) = \frac{(t_{i+1}-t)dT^s(t_i)+(t-t_i)dT^s(t_{i+1})}{t_{i+1}-t_i} - 2 \frac{\mathbf{r}^s(t)^T \mathbf{v}^s(t)}{c^2} \quad (\text{E.4.37})$$

where  $\mathbf{r}^s(t)$  and  $\mathbf{v}^s(t)$  are the satellite position and velocity derived from the precise ephemerides.

The precise ephemerides and clocks are applied in case that the processing option "Satellite Ephemeris/Clock" to "Precise".

#### (7) RTCM SSR orbit and clock corrections [18]

The RTCM SSR orbit and clock corrections are defined as the following parameters.

$$\Delta_{ssr}(t_0, IOD) = (\delta\theta_{radial}, \delta\theta_{along}, \delta\theta_{cross}, \delta\dot{\theta}_{radial}, \delta\dot{\theta}_{along}, \delta\dot{\theta}_{cross}, C_0, C_1, C_2) \quad (\text{E.4.41})$$

The  $IOD$  indicates the target broadcast ephemeris and clock parameters. The corrected satellite position

$\mathbf{r}^s(t)$  at time  $t$  is computed as:

$$\mathbf{e}_{\text{along}} = \frac{\mathbf{v}^s_{\text{broadcast}}(t)}{|\mathbf{v}^s_{\text{broadcast}}(t)|} \quad (\text{E.4.42})$$

$$\mathbf{e}_{\text{cross}} = \frac{\mathbf{r}^s_{\text{broadcast}}(t) \times \mathbf{v}^s_{\text{broadcast}}(t)}{|\mathbf{r}^s_{\text{broadcast}}(t) \times \mathbf{v}^s_{\text{broadcast}}(t)|} \quad (\text{E.4.43})$$

$$\mathbf{e}_{\text{radial}} = \mathbf{e}_{\text{along}} \times \mathbf{e}_{\text{cross}} \quad (\text{E.4.44})$$

$$\delta \mathbf{o} = \begin{pmatrix} \delta O_{\text{radial}} \\ \delta O_{\text{along}} \\ \delta O_{\text{cross}} \end{pmatrix} + \begin{pmatrix} \delta \dot{O}_{\text{radial}} \\ \delta \dot{O}_{\text{along}} \\ \delta \dot{O}_{\text{cross}} \end{pmatrix} (t - t_0) \quad (\text{E.4.43})$$

$$\mathbf{r}^s(t) = \mathbf{r}^s_{\text{broadcast}}(t) + (\mathbf{e}_{\text{radial}}, \mathbf{e}_{\text{along}}, \mathbf{e}_{\text{cross}}) \delta \mathbf{o} \quad (\text{E.4.45})$$

where:

$\mathbf{r}^s_{\text{broadcast}}(t)$  : satellite position at time  $t$  computed by the broadcast ephemeris with *IOD* (m)

$\mathbf{v}^s_{\text{broadcast}}(t)$  : satellite velocity at time  $t$  computed by the broadcast ephemeris with *IOD* (m/s)

The satellite velocity  $\mathbf{v}^s_{\text{broadcast}}(t)$  is computed by the following differential approximation with  $\Delta t = 0.001$  (s).

$$\mathbf{v}^s_{\text{broadcast}}(t) = \frac{\mathbf{r}^s_{\text{broadcast}}(t + \Delta t) - \mathbf{r}^s_{\text{broadcast}}(t)}{\Delta t} \quad (\text{E.4.46})$$

The corrected satellite clock bias  $dT^s(t)$  at the time  $t$  is also computed with the SSR correction as:

$$\delta C = C_0 + C_1(t - t_0) + C_2(t - t_0)^2 \quad (\text{E.4.47})$$

$$dT^s(t) = dT^s_{\text{broadcast}}(t) + \frac{\delta C}{C} \quad (\text{E.4.48})$$

where:

$dT^s_{\text{broadcast}}(t)$  : satellite clock bias at time  $t$  computed by the broadcast clock with *IOD* with the following relativity correction.

$$\Delta t_{\text{rel}} = -\frac{2\mathbf{r}^s_{\text{broadcast}}(t)^T \mathbf{v}^s_{\text{broadcast}}(t)}{C^2} \quad (\text{E.4.49})$$

The SSR corrections with broadcast ephemerides and clocks are applied in case that the processing option "Satellite Ephemeris/Clock" to "Broadcast+SSR APC" or "Broadcast+SSR CoM".

## E.5 Troposphere and Ionosphere Models

RTKLIB supports the following troposphere and ionosphere models.

### (1) Troposphere model

The standard atmosphere can be expressed as:

$$p = 1013.25 \times (1 - 2.2557 \times 10^{-5}h)^{5.2568} \quad (\text{E.5.1})$$

$$T = 15.0 - 6.5 \times 10^{-3}h + 273.15 \quad (\text{E.5.2})$$

$$e = 6.108 \times \exp\left\{\frac{17.15T - 4684.0}{T - 38.45}\right\} \times \frac{h_{rel}}{100} \quad (\text{E.5.3})$$

where  $p$  is the total pressure (hPa),  $T$  is the absolute temperature (K) of the air,  $h$  is the geodetic height above MSL (mean sea level),  $e$  is the partial pressure (hPa) of water vapor and  $h_{rel}$  is the relative humidity. The tropospheric delay  $T_r^s$  is expressed by the Saastamoinen model with  $p$ ,  $T$  and  $e$  derived from the standard atmosphere.

$$T_r^s = \frac{0.002277}{\cos z} \left\{ p + \left( \frac{1255}{T} + 0.05 \right) e - \tan^2 z \right\} \quad (\text{E.5.4})$$

where  $z$  is the zenith angle (rad) as  $z = \pi/2 - El_r^s$ .

The standard atmosphere and the Saastamoinen model are applied in case that the processing option "Troposphere Correction" is set to "Saastamoinen", where the geodetic height is approximated by the ellipsoidal height and the relative humidity is fixed to 70 %.

### (2) SBAS troposphere model

If the processing option "Troposphere correction" is set to "SBAS", the SBAS troposphere models defined in the SBAS receiver specifications are applied. The model often called as "MOPS model". Refer [8] A.4.2.4 for details.

### (3) Precise troposphere model

If the processing option "Troposphere Correction" is set to "Estimate ZTD" or "Estimate ZTD+Grad", a more precise troposphere model is applied with strict mapping functions as:

$$m(El_r^s) = m_W(El_r^s) \{ 1 + \cot E l_r^s (G_{N,r} \cos A z_r^s + G_{E,r} \sin A z_r^s) \} \quad (\text{E.5.5})$$

$$T_r^s = m_H(El_r^s) Z_{H,r} + m(El_r^s)(Z_{T,r} - Z_{H,r}) \quad (\text{E.5.6})$$

where:

- $Z_{T,r}$  : tropospheric zenith total delay (m)
- $Z_{H,r}$  : tropospheric zenith hydro-static delay (m)
- $m_H(El)$  : hydro-static mapping function
- $m_W(El)$  : wet mapping function

In RTKLIB, the tropospheric zenith hydro-static delay is given by Saastamoinen model (E.5.4) with the zenith angle  $z = 0$  and relative humidity  $h_{rel} = 0$ . For the mapping function, RTKLIB employs NMF (Niell mapping function)<sup>[70]</sup> as default. The zenith total delay  $Z_{T,r}$  and the gradient parameters  $G_{N,r}$ ,  $G_{H,r}$  (in the case of "Estimate ZTD+Grad") are estimated as unknown parameters in the parameter estimation process. For the mapping function, RTKLIB can uses GMF<sup>[71]</sup> by setting the compiler option -DIERS\_MODEL since ver. 2.4.2.

#### (4) Broadcast ionosphere model

For ionosphere correction for single frequency GNSS users, GPS and QZSS navigation data include the following broadcast ionospheric parameters.

$$\mathbf{p}_{ion} = (\alpha_0, \alpha_1, \alpha_2, \alpha_3, \beta_0, \beta_1, \beta_2, \beta_3)^T \quad (E.5.5)$$

By using these ionospheric parameters, the  $L_1$  ionospheric delay  $I_r^s$  (m) can be derived the following procedure<sup>[1]</sup>. The model is often called as Klobuchar model.

$$\psi = 0.0137/(El + 0.11) - 0.022 \quad (E.5.6)$$

$$\varphi_i = \varphi + \psi \cos Az \quad (E.5.7)$$

$$\lambda_i = \lambda + \psi \sin Az / \cos \varphi_i \quad (E.5.8)$$

$$\varphi_m = \varphi_i + 0.064 \cos(\lambda_i - 1.617) \quad (E.5.9)$$

$$t = 4.32 \times 10^4 \lambda_i + t \quad (E.5.10)$$

$$F = 1.0 + 16.0 \times (0.53 - El)^3 \quad (E.5.11)$$

$$x = 2\pi(t - 50400) / \sum_{n=0}^3 \beta_n \varphi_m^n \quad (E.5.12)$$

$$I_r^s = \begin{cases} F \times 5 \times 10^{-9} & (|x| > 1.57) \\ F \times \left(5 \times 10^{-9} + \sum_{n=1}^4 \alpha_n \varphi_m^n \times \left(1 - \frac{x^2}{2} + \frac{x^4}{24}\right)\right) & (|x| \leq 1.57) \end{cases} \quad (E.5.13)$$

Corrections by the broadcast ionosphere model are applied if the processing option "Ionosphere Correction" is set to "Broadcast" or "QZSS Broadcast".

#### (5) SBAS ionosphere model

SBAS corrections for ionospheric delay is provided by the message type 18 (ionospheric grid point

masks) and the message type 26 (ionospheric delay corrections). RTKLIB uses the SBAS ionospheric correction if the processing option "Ionosphere Correction" is set to "SBAS" and these SBAS messages are provided in an input file. For the algorithms for the model and the definition of IGPs (ionospheric grid points), refer A.4.4.9 and A.4.4.10 of the SBAS receiver specifications [8].

#### (6) Single layer model

The ionosphere is often modeled as a simple single layer model shown in Figure E.5-1. The single layer model is also called as a thin-shell model.

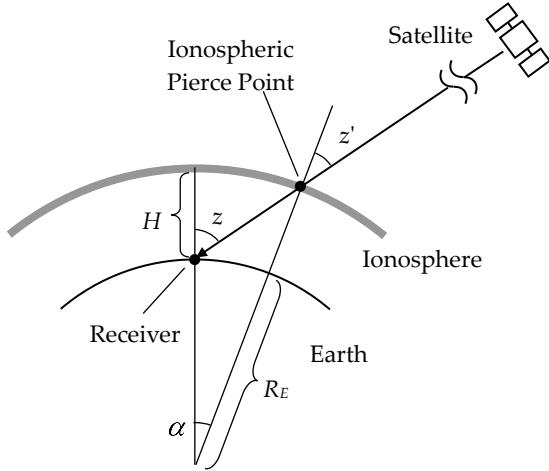


Figure E.5-1 Single Layer Ionosphere Model

In the model, the latitude  $\phi_{IPP}$  (rad) and the longitude  $\lambda_{IPP}$  (rad) of the IPP (ionospheric pierce point) can be derived from:

$$z = \pi/2 - El_r^s \quad (\text{E.5.14})$$

$$z' = \arcsin\left(\frac{R_E}{R_E+H} \sin z\right) \quad (\text{E.5.15})$$

$$\alpha = z - z' \quad (\text{E.5.16})$$

$$\phi_{IPP} = \arcsin(\cos \alpha \sin \phi_r + \sin \alpha \cos \phi_r \cos A z_r^s) \quad (\text{E.5.17})$$

$(\phi_r > 70^\circ \text{ and } \tan \alpha \cos A z_r^s > \tan(\pi/2 - \phi_r)) \text{ or } (\phi_r < -70^\circ \text{ and } -\tan \alpha \cos A z_r^s > \tan(\pi/2 + \phi_r))$

$$\lambda_{IPP} = \lambda_r + \pi - \arcsin \frac{\sin \alpha \sin A z_r^s}{\cos \phi_{IPP}} \quad (\text{E.5.18a})$$

(otherwise)

$$\lambda_{IPP} = \lambda_r + \arcsin \frac{\sin \alpha \sin A z_r^s}{\cos \phi_{IPP}} \quad (\text{E.5.18b})$$

where  $R_E$  is the radius of the earth (m) and  $H$  is the height of the ionosphere shell (m). RTKLIB usually

uses the values  $R_E = 6378137$  and  $H = 350000$ . Note that the earth surface or the ionosphere shell are approximated as a sphere in this model.

If the VTEC (vertical total electron content) value  $TEC(t, \phi_{IPP}, \lambda_{IPP})$  at the IPPP and the time  $t$  is given, the  $L_i$  ionospheric delay  $I_{r,i}^s$  (m) can be expressed as:

$$I_{r,i}^s = \frac{1}{\cos z'} \frac{40.3 \times 10^{16}}{f_i} TEC(t, \phi_{IPP}, \lambda_{IPP}) \quad (\text{E.5.19})$$

where  $f_i$  is the carrier frequency of signals (Hz).

The VTEC values are provided in several formats or equations. RTKLIB currently only supports the VTEC values provided by the IONEX format [24]. In the IONEX format, the VTEC are expressed as the point values in a latitude and longitude grid. RTKLIB interpolates these grid point values in the IONEX data to an appropriate IPP position by a simple bi-linear interpolation. These VTEC values are provided at every epoch time intervals in the IONEX file. The time interpolation should also be applied in the sun-fixed coordinates as:

$$TEC(t, \phi_{IPP}, \lambda_{IPP}) = \frac{(t-t_i)TEC(t_i, \phi_{IPP}, \lambda_{IPP} + \omega(t-t_i)) + (t_{i+1}-t)TEC(t_{i+1}, \phi_{IPP}, \lambda_{IPP} + \omega(t-t_{i+1}))}{t_{i+1}-t_i} \quad (\text{E.5.20})$$

where  $t_i$  and  $t_{i+1}$  ( $t_i \leq t < t_{i+1}$ ) are the time for the provided TEC data and  $\omega = 2\pi/86400$  is the rotation velocity of the sun to the earth. Correction by the single layer model with IONEX data is applied if the processing option "Ionosphere Correction" is set to "IONEX TEC" and IONEX data are provided as input files only in the post-processing mode.

#### (7) Ionosphere-free LC (linear combination)

To eliminate the ionosphere effects in the GNSS signal measurements, a LC (linear combination) of dual-frequency measurements is often utilized in GNSS data processing. The ionosphere-free LC of  $L_i$  and  $L_j$  pseudorange and phase-range are expressed as:

$$P_{r,LC}^s = C_i P_{r,i}^s + C_j P_{r,j}^s \quad (\text{E.5.21})$$

$$\Phi_{r,LC}^s = C_i \Phi_{r,i}^s + C_j \Phi_{r,j}^s \quad (\text{E.5.22})$$

where  $C_i$  and  $C_j$  are the coefficients of the ionosphere free LC. The  $C_i$  and  $C_j$  are derived from:

$$C_i = \frac{f_i^2}{f_i^2 - f_j^2} \quad (E.5.23)$$

$$C_j = \frac{-f_j^2}{f_i^2 - f_j^2} \quad (E.5.24)$$

where  $f_i$  and  $f_j$  are the frequencies (Hz) of  $L_i$  and  $L_j$  measurements. Current version RTKLIB always uses  $L_1$  and  $L_2$  for GPS, GLONASS and QZSS,  $L_1$  and  $L_5$  for Galileo for the ionosphere-free LC. If setting the processing option "Ionosphere Correction" to "Iono-Free LC" in the Single or PPP modes, the ionosphere-free LC is used for basic measurements to eliminate the ionosphere term. Note that the ionosphere-free LC model is not applied for the Kinematic, Static or Moving-base modes. Refer the E.7 (6) for details.

## E.6 Single Point Positioning

RTKLIB employs an iterated weighted LSE (least square estimation) for the "Single" (single point positioning) mode with or without SBAS corrections.

### (1) Linear LSE

Assume a measurement vector  $\mathbf{y}$  are given and it can be modeled as the following linear equations of an unknown parameter vector  $\mathbf{x}$  and a random measurement error vector  $\mathbf{v}$ .

$$\mathbf{y} = \mathbf{Hx} + \mathbf{v} \quad (\text{E.6.1})$$

The least square cost function  $J_{LS}$  is defined as the sum of the squared measurement errors as:

$$J_{LS} = v_1^2 + v_2^2 + \dots + v_m^2 = \mathbf{v}^T \mathbf{v} \quad (\text{E.6.2})$$

By using (E.6.1) and (E.6.2), the cost function can be rewritten as:

$$\begin{aligned} J_{LS} &= (\mathbf{y} - \mathbf{Hx})^T (\mathbf{y} - \mathbf{Hx}) \\ &= \mathbf{y}^T \mathbf{y} - \mathbf{y}^T \mathbf{Hx} - \mathbf{x}^T \mathbf{H}^T \mathbf{y} + \mathbf{x}^T \mathbf{H}^T \mathbf{Hx} \end{aligned} \quad (\text{E.6.3})$$

To minimize the cost function, the gradient of  $J_{LS}$  should be zero. Then

$$\begin{aligned} \frac{\partial J_{LS}}{\partial \mathbf{x}} &= \mathbf{0}^T - \mathbf{y}^T \mathbf{H} - (\mathbf{H}^T \mathbf{y})^T + (\mathbf{H}^T \mathbf{Hx})^T + \mathbf{x}^T \mathbf{H}^T \mathbf{H} \\ &= -2\mathbf{y}^T \mathbf{H} + 2\mathbf{x}^T \mathbf{H}^T \mathbf{H} = \mathbf{0} \end{aligned} \quad (\text{E.6.4})$$

It gives so called a "normal equation" as:

$$\mathbf{H}^T \mathbf{Hx} = \mathbf{H}^T \mathbf{y} \quad (\text{E.6.5})$$

To solve the normal equation, we can get the estimated unknown parameter vector  $\hat{\mathbf{x}}$  by the LSE as:

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{y} \quad (\text{E.6.6})$$

If the weights of each measurements are given, the cost function (E.6.3) can be rewritten by using a

weight matrix  $\mathbf{W}$ .

$$J_{WLS} = \mathbf{v}^T \mathbf{W} \mathbf{v} \quad (\text{E.6.7})$$

To minimize the cost function  $J_{WLS}$ , we can obtain the estimated unknown parameter vector by the weighted LSE by the similar way for the simple LSE as:

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} \mathbf{y} \quad (\text{E.6.8})$$

The weight matrix  $\mathbf{W}$  for the weighted LSE is often given as:

$$\mathbf{W} = \text{diag}(\sigma_1^{-2}, \sigma_2^{-2}, \dots, \sigma_m^{-2})$$

where  $\sigma_i$  is the a-priori standard deviation of the i-th measurement error.

## (2) Gauss-Newton iteration for non-linear LSE

In case that the measurements are not given as linear models, the measurement equations can be written by a general non-linear vector function as:

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v} \quad (\text{E.6.9})$$

where  $\mathbf{h}(\mathbf{x})$  is a measurement vector function of a parameters vector  $\mathbf{x}$ . The equation can be extended by using Taylor series around an initial parameter vector  $\mathbf{x}_0$  as:

$$\mathbf{h}(\mathbf{x}) = \mathbf{h}(\mathbf{x}_0) + \mathbf{H}(\mathbf{x} - \mathbf{x}_0) + .. \quad (\text{E.6.10})$$

where  $\mathbf{H}$  is a partial derivatives matrix of  $\mathbf{h}(\mathbf{x})$  with respect to  $\mathbf{x}$  at  $\mathbf{x} = \mathbf{x}_0$ :

$$\mathbf{H} = \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \Big|_{\mathbf{x}=\mathbf{x}_0} \quad (\text{E.6.11})$$

Assume the initial parameters are adequately near the true values and the second and further terms of the Taylor series can be neglected. We can approximate (E.6.9) as:

$$\mathbf{y} \approx \mathbf{h}(\mathbf{x}_0) + \mathbf{H}(\mathbf{x} - \mathbf{x}_0) + \mathbf{v} \quad (\text{E.6.12})$$

Then we can obtain the following linear equation.

$$\mathbf{y} - \mathbf{h}(\mathbf{x}_0) = \mathbf{H}(\mathbf{x} - \mathbf{x}_0) + \mathbf{v} \quad (\text{E.6.13})$$

By applying linear weighted LSE (E.6.8) for (E.6.13), we can get the normal equation for non-linear weighted LSE:

$$\mathbf{H}^T \mathbf{W} \mathbf{H} (\hat{\mathbf{x}} - \mathbf{x}_0) = \mathbf{H}^T \mathbf{W} (\mathbf{y} - \mathbf{h}(\mathbf{x}_0)) \quad (\text{E.6.14})$$

So we can obtain the estimated unknown parameter vector  $\hat{\mathbf{x}}$  by:

$$\hat{\mathbf{x}} = \mathbf{x}_0 + (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} (\mathbf{y} - \mathbf{h}(\mathbf{x}_0)) \quad (\text{E.6.15})$$

If the initial parameters  $\mathbf{x}_0$  are not enough near the true values, we can iteratively improve the estimated parameters like:

$$\hat{\mathbf{x}}_0 = \mathbf{x}_0 \quad (\text{E.6.16})$$

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} (\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}}_i)) \quad (\text{E.6.17})$$

If the iteration is converged, we can obtain the final estimated parameters as:

$$\hat{\mathbf{x}} = \lim_{i \rightarrow \infty} \hat{\mathbf{x}}_i \quad (\text{E.6.18})$$

The iterated LSE is often called as Gauss-Newton method. Note that such the iterations are not always converged by the simple Gauss-Newton method especially for ill-conditioned measurement equations having large non-linearity. In these cases, we should employ another strategy for such non-linear LSE. The most popular way for the non-linear LSE is LM (Levenberg-Marquardt) method [69].

### (3) Estimation of receiver position and clock bias

For "Single" mode as "Positioning Mode", the following single-point positioning procedure is applied to obtain the final solution by epoch-by-epoch basis. For an epoch time, the unknown parameters vector  $\mathbf{x}$  is defined as:

$$\mathbf{x} = (\mathbf{r}_r^T, cdt_r)^T \quad (\text{E.6.19})$$

The pseudorange measurement vector  $\mathbf{y}$  can be given as:

$$\mathbf{y} = (P_r^1, P_r^2, P_r^3, \dots, P_r^m)^T \quad (\text{E.6.20})$$

where  $P_r^s$  is the pseudorange measurement. If the processing option "Ionosphere Correction" set to "Iono-Free LC", the ionosphere-free LC (linear combination) pseudorange defined in Appendix E.5 (7) is used. In other cases, just the  $L_1$  pseudorange is used.

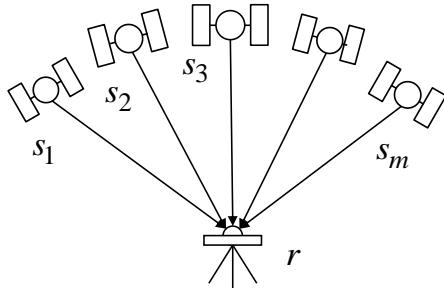


Figure E.6-1 Satellite Geometry for Single Point Positioning

The measurement equation and its partial derivative matrix for the single point positioning are formed as:

$$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} \rho_r^1 + cdt_r - cdT^1 + I_r^1 + T_r^1 \\ \rho_r^2 + cdt_r - cdT^2 + I_r^2 + T_r^2 \\ \rho_r^3 + cdt_r - cdT^3 + I_r^3 + T_r^{s3} \\ \vdots \\ \rho_r^m + cdt_r - cdT^m + I_r^m + T_r^m \end{pmatrix} \mathbf{H} = \begin{pmatrix} -\mathbf{e}_r^{1T} & 1 \\ -\mathbf{e}_r^{2T} & 1 \\ -\mathbf{e}_r^{3T} & 1 \\ \vdots & \vdots \\ -\mathbf{e}_r^{mT} & 1 \end{pmatrix} \quad (\text{E.6.21})$$

where the geometric range  $\rho_r^s$  and LOS vector  $\mathbf{e}_r^s$  are given by E.3 (4) and E.3 (5) with the satellite and receiver positions. The satellite positions  $\mathbf{r}^s$  and the clock biases  $dT^s$  are also derived from the GNSS satellite ephemerides and clocks described in E.4 according to the processing option "Satellite Ephemeris/Clock".

To solve the measurement equation to obtain the final estimated receiver position and the receiver clock bias, RTKLIB employs the iterated weighted LSE as:

$$\hat{\mathbf{x}}_{i+1} = \hat{\mathbf{x}}_i + (\mathbf{H}^T \mathbf{W} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{W} (\mathbf{y} - \mathbf{h}(\hat{\mathbf{x}}_i)) \quad (\text{E.6.22})$$

For the initial parameter vector  $\mathbf{x}_0$  for the iterated weighted LSE, just all 0 are used for the first epoch of the single point positioning. Once a solution obtained, the position is used for the next epoch initial receiver position. For the weight matrix  $\mathbf{W}$ , RTKLIB uses the following formulas

$$\mathbf{W} = \text{diag}(\sigma_1^{-2}, \sigma_2^{-2}, \dots, \sigma_m^{-2}) \quad (\text{E.6.23})$$

$$\sigma^2 = F^s R_r (a_\sigma^{-2} + b_\sigma^{-2} / \sin E l_r^s) + \sigma_{eph}^{-2} + \sigma_{ion}^{-2} + \sigma_{trop}^{-2} + \sigma_{bias}^{-2} \quad (\text{E.6.24})$$

where:

$F^s$  : satellite system error factor

(1:GPS, Galileo, QZSS and BeiDou, 1.5: GLONASS, 3.0: SBAS)

$R_r$  : code/carrier-phase error ratio

$a_\sigma, b_\sigma$  : carrier-phase error factor  $a$  and  $b$  (m)

$\sigma_{eph}$  : standard deviation of ephemeris and clock error (m)

$\sigma_{ion}$  : standard deviation of ionosphere correction model error (m)

$\sigma_{trop}$  : standard deviation of troposphere correction model error (m)

$\sigma_{bias}$  : standard deviation of code bias error (m)

For the standard deviation of ephemeris and clock error, URA (user range accuracy) or similar indicators are used in RTKLIB. By several iterations, the solution is converged in the normal case and the estimated receiver position  $\hat{\mathbf{r}}_r$  and the receiver clock bias  $d\hat{t}_r$  are obtained.

$$\hat{\mathbf{x}} = \lim_{i \rightarrow \infty} \hat{\mathbf{x}}_i = (\hat{\mathbf{r}}_r^T, c\hat{d}t_r)^T \quad (\text{E.6.25})$$

The estimated receiver clock bias  $d\hat{t}_r$  is not explicitly output to the solution file. Instead, it is incorporated in the solution time-tag. That means the solution time-tag indicates not the receiver time-tag but the true signal reception time measured in GPST.

#### (4) Estimation of receiver velocity and clock-drift

If Doppler frequency measurements of GNSS signals are given, receiver velocities and clock-drifts can be estimated the following procedure. For an epoch time, the unknown parameters vector  $\mathbf{x}$  for the velocity estimation is defined as:

$$\mathbf{x} = (\mathbf{v}_r^T, cdt_r)^T \quad (\text{E.6.26})$$

where  $\mathbf{v}_r$  and  $dt_r$  are the receiver velocity in ECEF (m/s) and the receiver clock-drift (s/s), respectively.

The range-rate measurement vector  $\mathbf{y}$  can be given as:

$$\mathbf{y} = (-\lambda_i D_{r,i}^1, -\lambda_i D_{r,i}^2, -\lambda_i D_{r,i}^3, \dots, -\lambda_i D_{r,i}^m)^T \quad (\text{E.6.27})$$

where  $D_{r,i}^s$  is the  $L_i$  Doppler frequency measurement of the satellite  $s$ . RTKLIB always uses  $L_1$  Doppler frequency measurements. These measurement equations and its partial derivative matrix are formed as:

$$\mathbf{h}(\mathbf{x}) = \begin{pmatrix} r_r^1 + cdt_r - cd\dot{T}^1 \\ r_r^2 + cdt_r - cd\dot{T}^2 \\ r_r^3 + cdt_r - cd\dot{T}^3 \\ \vdots \\ r_r^m + cdt_r - cd\dot{T}^m \end{pmatrix} \mathbf{H} = \begin{pmatrix} -\mathbf{e}_r^{1T} & 1 \\ -\mathbf{e}_r^{2T} & 1 \\ -\mathbf{e}_r^{3T} & 1 \\ \vdots & \vdots \\ -\mathbf{e}_r^{mT} & 1 \end{pmatrix} \quad (\text{E.6.28})$$

The range-rate  $r_r^s$  between the receiver and the satellite in these equations is derived from:

$$r_r^s = \mathbf{e}_r^{sT}(\mathbf{v}^s(t^s) - \mathbf{v}_r) + \frac{\omega_e}{c}(v_y^s x_r + y^s v_{x,r} - v_x^s y_r - x^s v_{y,r}) \quad (\text{E.6.29})$$

where  $\mathbf{v}^s = (v_x^s, v_y^s, v_z^s)^T$  and  $\mathbf{v}_r = (v_{x,r}, v_{y,r}, v_{z,r})^T$ . By using the iterated LSE similar to the estimation of the receiver position, we can obtain the receiver velocity and clock-drift as:

$$\hat{\mathbf{x}} = \lim_{i \rightarrow \infty} \hat{\mathbf{x}}_i = (\dot{\mathbf{v}}_r^T, c\hat{t}_r)^T \quad (\text{E.6.30})$$

where the weight matrix  $\mathbf{W}$  is set to  $\mathbf{I}$  (non-weighted LSE).

## (5) Solution validation and RAIM FDE

The estimated receiver positions described in (3) might include invalid solutions due to unmodeled measurement errors. To test whether the valid solution or not and reject the invalid solutions, RTKLIB applies the following validation after obtaining the receiver position estimated. If the validation failed, the solution is rejected with warning messages.

$$v_s = \frac{(P_r^s - (\hat{P}_r^s + c\hat{t}_r - cdT^s + l_r^s + T_r^s))}{\sigma_s} \quad (\text{E.6.31})$$

$$\mathbf{v} = (v_1, v_2, v_3, \dots, v_m)^T \quad (\text{E.6.32})$$

$$\frac{\mathbf{v}^T \mathbf{v}}{m-n-1} < \chi_\alpha^2(m-n-1) \quad (\text{E.6.33})$$

$$GDOP < GDOP_{thres} \quad (\text{E.6.34})$$

where  $n$  is the number of estimated parameters and  $m$  is the number of measurements.  $\chi_\alpha^2(n)$  is chi-square distribution of the degree of freedom  $n$  and  $\alpha = 0.001$  (0.1%).  $GDOP$  is geometric DOP (dilution

of precision).  $GDOP_{thres}$  can be set as the processing option "Reject Threshold of GDOP".

In addition to the solution validation described above, RAIM (receiver autonomous integrity monitoring) FDE (fault detection and exclusion) function is added in ver.2.4.2. If the processing option "RAIM FDE" is enabled and the chi-squared test by (E.6.33) is failed, RTKLIB retries the estimation by excluding one by one of the visible satellites. After all of retries, the estimated receiver position with the minimum normalized squared residuals  $\mathbf{v}^T \mathbf{v}$  is selected as the final solution. In such scheme, an invalid measurement, which might be due to satellite malfunction, receiver fault or large multipath, is excluded as an outlier. Note that this feature does not effective with two or more invalid measurements. It also needs two redundant visible satellites, that means at least 6 visible satellites are necessary to obtain the final solution.

## E.7 Kinematic, Static and Moving-Baseline

RTKLIB employs EKF (extended Kalman filter) in order to obtain the final solutions in DGPS/DGNSS, Static, Kinematic and Moving-baseline modes in conjunction with the GNSS signal measurement models in Appendix E.3 and the troposphere and ionosphere models in Appendix E.5.

### (1) EKF formulation [65]

By using EKF, a state vector  $\mathbf{x}$  for unknown model parameters and its covariance matrix  $P$  can be estimated with a measurement vector  $y_k$  at an epoch  $t_k$  by:

$$\hat{\mathbf{x}}_k(+) = \hat{\mathbf{x}}_k(-) + \mathbf{K}_k(\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_k(-))) \quad (\text{E.7.1})$$

$$\mathbf{P}_k(+) = (\mathbf{I} - \mathbf{K}_k \mathbf{H}(\hat{\mathbf{x}}_k(-))) \mathbf{P}_k(-) \quad (\text{E.7.2})$$

$$\mathbf{K}_k = \mathbf{P}_k(-) \mathbf{H}(\hat{\mathbf{x}}_k(-)) (\mathbf{H}(\hat{\mathbf{x}}_k(-)) \mathbf{P}_k(-) \mathbf{H}(\hat{\mathbf{x}}_k(-))^T + \mathbf{R}_k)^{-1} \quad (\text{E.7.3})$$

where  $\hat{\mathbf{x}}_k$  and  $\mathbf{P}_k$  are the estimated state vector and its covariance matrix at the epoch time  $t_k$ . (-) and (+) indicates before and after measurement update of EKF.  $h(x)$ ,  $H(x)$  and  $R_k$  are the measurements model vector, the matrix of partial derivatives and the covariance matrix of measurement errors, respectively. Assuming the system-model linear, the time update of the state vector and its covariance matrix for EKF is expressed as:

$$\hat{\mathbf{x}}_{k+1}(-) = \mathbf{F}_k^{k+1} \hat{\mathbf{x}}_k(+) \quad (\text{E.7.4})$$

$$\mathbf{P}_{k+1}(-) = \mathbf{F}_k^{k+1} \mathbf{P}_k(+) \mathbf{F}_k^{k+1T} + \mathbf{Q}_k^{k+1} \quad (\text{E.7.5})$$

where  $F_k^{k+1}$  and  $Q_k^{k+1}$  are the transition matrix and the covariance matrix of the system noise from epoch time  $t_k$  to  $t_{k+1}$ .

### (2) DD (double-difference) measurement model for short baseline

For carrier-based relative positioning with a short length (< 10 km) baseline between the rover  $r$  and the base-station  $b$ , the following DD measurement equations are generally used for the  $L_i$  phase-range and pseudorange. In these equations, the satellite and receiver clock biases, and the ionospheric and tropospheric effects and other minor correction terms are almost eliminated by using DD technique.

$$\begin{aligned} \phi_{rb,i}^{jk} &= \rho_{rb}^{jk} + \lambda_i(B_{rb,i}^j - B_{rb,i}^k) + d\Phi_{r,i}^s + \varepsilon_\phi \\ P_{rb,i}^{jk} &= \rho_{rb}^{jk} + \varepsilon_p \end{aligned} \quad (\text{E.7.6})$$

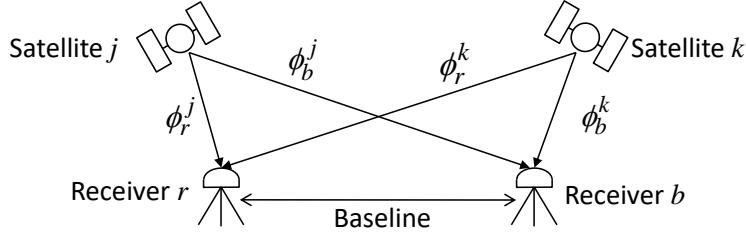


Figure E.7-1 DD (double-difference) Formulation

where the  $d\Phi_{r,i}^s$  is the carrier-phase correction terms, which can be neglected in the short-baseline case except for the receiver PCV terms with different antennas. To obtain the geometric range  $\rho_r^s$  in the equation, the base-station position  $\mathbf{r}_b$  is fixed to predetermined values except for the moving-baseline case.

Note that the SD between receivers had better to be made between the measurements with the same epoch time. However, the receivers are not perfectly synchronized due to the different receiver clock biases. In some typical cases, the sampling interval of the rover is different from the base station like 10 Hz and 1 Hz. To control the SD, RTKLIB takes a simple criterion to select a measurement pair. RTKLIB simply selects the last measurement before or equal to the epoch time of the rover measurement. The epoch time difference between the rover and the base station is sometimes called as "Age of Differential". As the time difference grows, the accuracy of the solution is gradually degraded due to the satellite clock drift and the variation of ionosphere delay. To compensate only the satellite clock drift, RTKLIB corrects the SD measurement by using broadcast SV clock parameters. The maximum "Age of Differential" is set as the processing option "MAX Age of Diff".

As to the satellite-side SD generation, RTKLIB selects a reference satellite with the maximum elevation angle on the epoch-by-epoch basis. Note that no satellite-side SD is generated between satellites of different navigation systems like between GPS and GLONASS. It is because that the receiver usually has different group delays for the signal of different navigation system even if they have the same carrier frequency . The group delay difference in receivers is called as a receiver ISB (inter system bias).

Assuming the use of triple-frequency GPS/GNSS receivers for both of the rover and the base-station, the unknown state vector  $\mathbf{x}$  to be estimated can be defined as:

$$\mathbf{x} = (\mathbf{r}_r^T, \mathbf{v}_r^T, \mathbf{B}_1^T, \mathbf{B}_2^T, \mathbf{B}_5^T)^T \quad (\text{E.7.7})$$

where  $\mathbf{B}_i = (B_{rb,i}^1, B_{rb,i}^2, \dots, B_{rb,i}^m)^T$  is  $L_i$  SD (single-difference) carrier-phase biases (cycle). As the RTKLIB implementation, it internally uses SD carrier-phase biases instead of DD to avoid bothersome hand-over handling of reference satellites. The SD biases also assist to resolve integer ambiguities in GLONASS FDMA signals.

The measurement vector  $\mathbf{y}$  is also defined with DD phase-range and pseudorange measurements as:

$$\mathbf{y} = (\boldsymbol{\Phi}_1^T, \boldsymbol{\Phi}_2^T, \boldsymbol{\Phi}_5^T, \mathbf{P}_1^T, \mathbf{P}_2^T, \mathbf{P}_5^T)^T \quad (\text{E.7.8})$$

where:

$$\begin{aligned} \boldsymbol{\Phi}_i &= (\phi_{rb,i}^{12}, \phi_{rb,i}^{13}, \phi_{rb,i}^{14}, \dots, \phi_{rb,i}^{1m})^T \\ \mathbf{P}_i &= (P_{rb,i}^{12}, P_{rb,i}^{13}, P_{rb,i}^{14}, \dots, P_{rb,i}^{1m})^T \end{aligned}$$

### (3) Measurement update of EKF for short baseline

By using the equation (E.7.6), the measurement model vector  $h(x)$ , the matrix of partial derivatives  $H(x)$  and the covariance matrix of measurement errors  $\mathbf{R}$  can be written as:

$$\mathbf{h}(x) = (\mathbf{h}_{\phi,1}^T, \mathbf{h}_{\phi,2}^T, \mathbf{h}_{\phi,5}^T, \mathbf{h}_{P,1}^T, \mathbf{h}_{P,2}^T, \mathbf{h}_{P,5}^T)^T \quad (\text{E.7.9})$$

$$H(x) = \left. \frac{\partial \mathbf{h}(x)}{\partial x} \right|_{x=\hat{x}} = \begin{pmatrix} -\mathbf{D}\mathbf{E} & \mathbf{0} & \lambda_1 \mathbf{D} & \mathbf{0} & \mathbf{0} \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{0} & \lambda_2 \mathbf{D} & \mathbf{0} \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \lambda_5 \mathbf{D} \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \quad (\text{E.7.10})$$

$$\mathbf{R} = \begin{pmatrix} \mathbf{D}\mathbf{R}_{\phi,1}\mathbf{D}^T & & & & \\ & \mathbf{D}\mathbf{R}_{\phi,2}\mathbf{D}^T & & & \\ & & \mathbf{D}\mathbf{R}_{\phi,5}\mathbf{D}^T & & \\ & & & \mathbf{D}\mathbf{R}_{P,1}\mathbf{D}^T & \\ & & & & \mathbf{D}\mathbf{R}_{P,2}\mathbf{D}^T \\ & & & & & \mathbf{D}\mathbf{R}_{P,5}\mathbf{D}^T \end{pmatrix} \quad (\text{E.7.11})$$

where:

$$\mathbf{h}_{\phi,i} = \begin{pmatrix} \rho_{rb}^{12} + \lambda_i(B_{rb}^1 - B_{rb}^2) \\ \rho_{rb}^{13} + \lambda_i(B_{rb}^1 - B_{rb}^3) \\ \vdots \\ \rho_{rb}^{1m} + \lambda_i(B_{rb}^1 - B_{rb}^m) \end{pmatrix}, \mathbf{h}_{P,i} = \begin{pmatrix} \rho_{rb}^{12} \\ \rho_{rb}^{13} \\ \vdots \\ \rho_{rb}^{1m} \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 1 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \cdots & -1 \end{pmatrix} : \text{SD (single-differencing) matrix}$$

$$E =$$

$$\mathbf{R}_{\phi,i} = \text{diag}(2\sigma_{\phi,i}^1{}^2, 2\sigma_{\phi,i}^2{}^2, \dots, 2\sigma_{\phi,i}^m{}^2)$$

$$\mathbf{R}_{P,i} = \text{diag}(2\sigma_{P,i}^1{}^2, 2\sigma_{P,i}^2{}^2, \dots, 2\sigma_{P,i}^m{}^2)$$

$\sigma_{\phi,i}^s$  : standard deviation of  $L_i$  phase-range measurement error (m)

$\sigma_{P,i}^s$  : standard deviation of  $L_i$  pseudorange measurement error (m)

By solving the EKF formulas (E.7.1) with these equations, the estimated rover antenna position, velocity and float SD carrier-phase biases the epoch time  $t_k$  are obtained.

#### (4) Time update of EKF

For the kinematic positioning mode with receiver dynamics (Positioning Mode = Kinematic and REC Dynamics = ON) in RTKLIB, the time update of EKF (E.7.2) is expressed with:

$$\mathbf{F}_k^{k+1} = \begin{pmatrix} \mathbf{I}_{3 \times 3} & & \\ & \mathbf{I}_{3 \times 3} \tau_r & \\ & & \mathbf{I}_{(3m-3) \times (3m-3)} \end{pmatrix}, \mathbf{Q}_k^{k+1} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & & \\ & \mathbf{Q}_v & \\ & & \mathbf{0}_{(3m-3) \times (3m-3)} \end{pmatrix}$$

(E.7.12)

where:

$$\mathbf{Q}_v = \mathbf{E}_r^T \text{diag}(\sigma_{ve}^2 \tau_r, \sigma_{vn}^2 \tau_r, \sigma_{vu}^2 \tau_r) \mathbf{E}_r$$

and  $\tau_r = t_{k+1} - t_k$  is GPS/GNSS receiver sampling interval (s),  $(\sigma_{ve}, \sigma_{vn}, \sigma_{vu})$  are the standard deviations of east, north and up components of the rover velocity system noises (m/s/ $\sqrt{s}$ ).

For the pure kinematic mode without receiver dynamics (Positioning Mode = Kinematic and REC Dynamics = OFF), equations (E.7.9) is be replaced by:

$$\mathbf{F}_k^{k+1} = \begin{pmatrix} \mathbf{I}_{3 \times 3} & & \\ & \mathbf{I}_{3 \times 3} & \\ & & \mathbf{I}_{(3m-3) \times (3m-3)} \end{pmatrix}, \mathbf{Q}_k^{k+1} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & & \\ & \mathbf{0}_{3 \times 3} & \\ & & \mathbf{0}_{(3m-3) \times (3m-3)} \end{pmatrix}$$

(E.7.13)

To avoid numerical instability by adding infinite process noises to the variances of the receiver position, the receiver position states are instead reset to the initial guess values at every epochs and adequately large process noises ( $10^4$  m<sup>2</sup>) are added to the variance in RTKLIB. The initial position is derived from the single point positioning process which is used to avoid the iteration for non-linear signal measurement model.

In the static mode (Positioning Mode = Static), equations (E.7.10) is simply replaced by:

$$\mathbf{F}_k^{k+1} = \begin{pmatrix} \mathbf{I}_{3 \times 3} & & \\ & \mathbf{I}_{3 \times 3} & \\ & & \mathbf{I}_{(3m-3) \times (3m-3)} \end{pmatrix}, \mathbf{Q}_k^{k+1} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & & \\ & \mathbf{0}_{3 \times 3} & \\ & & \mathbf{0}_{(3m-3) \times (3m-3)} \end{pmatrix} \quad (\text{E.7.14})$$

In the instantaneous ambiguity resolution mode (Integer Ambiguity Resolution = Instantaneous), the time update of the SD carrier-phase biases  $\mathbf{B}_i$  are handled in a little different way from described above. In this mode, values of the carrier-phase bias states are not succeeded to the next epoch by the EKF time update. The biases are reset to the initial guess values at every epochs and adequately large process noises ( $10^4 \text{ m}^2$ ) are added to the variance. If a cycle-slip is detected in the measurement data, the state of corresponding SD carrier-phase bias is also reset to initial value. RTKLIB detects the cycle-slips by LLI (loss of lock indicator) in the input measurement data and geometry-free LC (linear combination) phase jumps if the dual frequency measurements are available. The cycle-slip threshold can be changed by the processing option "Slip Thres".

##### (5) Integer ambiguity resolution

Once the estimated states obtained in the EKF measurement update, the float carrier-phase ambiguities can be resolved into integer values in order to improve accuracy and convergence time (Integer Ambiguity Res = Continuous, Instantaneous or Fix and Hold). At first, the estimated states and their covariance matrix are transformed to DD forms by:

$$\hat{\mathbf{x}}'_k = \mathbf{G}\hat{\mathbf{x}}_k(+) = (\hat{\mathbf{r}}_r^T, \hat{\mathbf{v}}_r^T, \hat{\mathbf{N}}^T)^T \quad (\text{E.7.15})$$

$$\mathbf{P}'_k = \mathbf{G}\mathbf{P}_k(+) \mathbf{G}^T = \begin{pmatrix} \mathbf{Q}_R & \mathbf{Q}_{NR} \\ \mathbf{Q}_{RN} & \mathbf{Q}_N \end{pmatrix} \quad (\text{E.7.16})$$

where:

$$\mathbf{G} = \begin{pmatrix} \mathbf{I}_{6 \times 6} & & \\ & \mathbf{D} & \\ & & \mathbf{D} \\ & & & \mathbf{D} \end{pmatrix}: \text{SD to DD transformation matrix}$$

In this transformation, the SD carrier-phase biases are transferred to the DD carrier-phase form in order to eliminate receiver initial phase terms to obtain integer ambiguities  $\hat{\mathbf{N}}$  and their covariance  $\mathbf{Q}_N$ . In these formulas, the most appropriate integer vector  $\tilde{\mathbf{N}}$  for the integer ambiguities is obtained by solving an ILS (integer least square) problem expressed as:

$$\tilde{\mathbf{N}} = \underset{\mathbf{N} \in \mathbb{Z}}{\operatorname{argmin}} ((\mathbf{N} - \hat{\mathbf{N}})^T \mathbf{Q}_N^{-1} (\mathbf{N} - \hat{\mathbf{N}})) \quad (\text{E.7.17})$$

To solve the ILS problem, a well-known efficient search strategy LAMBDA [66] and its extension

MLAMBDA [67] are employed in RTKLIB. LAMBDA and MLAMBDA offer the combination of a linear transformation to shrink the integer vector search space and a skillful tree-search procedure in the transformed space. The integer vector solution by these procedures is validated by the following simple "Ratio-Test". In the "Ratio-Test", the ratio-factor  $R$ , defined as the ratio of the weighted sum of the squared residuals by the second best solution  $\bar{\mathbf{N}}_2$  to one by the best  $\bar{\mathbf{N}}$ , is used to check the reliability of the solution. The validation threshold  $R_{thres}$  can be set by the processing option "Min Ratio to Fix Ambiguity". Current version RTKLIB just only supports a fixed threshold value.

$$R = \frac{(\bar{\mathbf{N}}_2 - \hat{\mathbf{N}})^T \mathbf{Q}_N^{-1} (\bar{\mathbf{N}}_2 - \hat{\mathbf{N}})}{(\bar{\mathbf{N}} - \hat{\mathbf{N}})^T \mathbf{Q}_N^{-1} (\bar{\mathbf{N}} - \hat{\mathbf{N}})} > R_{thres}$$

(E.7.18)

After the validation, the "FIXED" solution of the rover antenna position and velocity  $\check{\mathbf{r}}_r$  and  $\check{\mathbf{v}}_r$  are obtained by solving the following equation. If the validation failed, RTKLIB outputs the "FLOAT" solution  $\hat{\mathbf{r}}_r$  and  $\hat{\mathbf{v}}_r$  instead.

$$\begin{pmatrix} \check{\mathbf{r}}_r \\ \check{\mathbf{v}}_r \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{r}}_r \\ \hat{\mathbf{v}}_r \end{pmatrix} - \mathbf{Q}_{RN} \mathbf{Q}_N^{-1} (\bar{\mathbf{N}} - \hat{\mathbf{N}}) \quad (E.7.19)$$

In case the processing option is set as the "Fix and Hold" mode (Integer Ambiguity Resolution = Fix and Hold) and the fixed solution properly validated by the previous test, the DD carrier-phase bias parameters are tightly constraint to the fixed integer values. For these purpose, RTKLIB inputs the following "pseudo" measurements to EKF and updates EKF by (F.7.1).

$$\mathbf{y} = \bar{\mathbf{N}} \quad (E.7.20)$$

$$\mathbf{h}(\mathbf{x}) = \mathbf{G}\mathbf{x} \quad (E.7.21)$$

$$\mathbf{H}(\mathbf{x}) = \mathbf{G} \quad (E.7.22)$$

$$\mathbf{R} = \text{diag}(\sigma_c^2, \sigma_c^2, \sigma_c^2, \dots) \quad (E.7.23)$$

where:

$$\mathbf{G} = \begin{pmatrix} \mathbf{0} & \mathbf{D} \\ \mathbf{0} & \mathbf{D} \\ \mathbf{0} & \mathbf{D} \end{pmatrix} : \text{SD to DD transformation matrix}$$

$\sigma_c$ : constraint to fixed integer ambiguities (= 0.001 cycle).

The "Fix and Hold" mode was firstly introduced in RTKLIB ver. 2.4.0 in order to improve the fixing ratio especially in the kinematic mode to tracking moving receivers.

(6) Long baseline DD measurement model

For the long baseline processing between the rover  $r$  and the base station  $b$ , the following DD measurement equations can be formed similar to the short baseline DD model.

$$\begin{aligned}\Phi_{rb,i}^{jk} &= \rho_{rb}^{jk} - I_{rb,k}^{jk} + T_{rb}^{jk} + \lambda_i(B_{rb,i}^j - B_{rb,i}^k) + d\phi_{r,i}^s + \varepsilon_\phi \\ P_{rb,i}^{jk} &= \rho_{rb}^{jk} + I_{rb,i}^{jk} + T_{rb}^{jk} + \varepsilon_p\end{aligned}\quad (\text{E.7.24})$$

where the terms  $I_{r,i}^s$  and  $T_r^s$  as the  $L_i$ ionosphere delay (m) and troposphere delay (m) are added to the short baseline DD model. Precise ephemerides for satellite positions should be used to mitigate the broadcast ephemeris error for the baseline over 100 km. In the carrier phase correction terms  $d\phi_{r,i}^s$ , the earth tides effects should be taken account for the longer baseline than 500 km. To eliminate the ionosphere terms, an ionosphere-free LC (linear combination) is sometimes formed. However, RTKLIB does not use such explicit LC but does directly estimate the ionosphere terms with dual or triple frequency measurements by EKF for baseline processing.

The unknown state vector  $x$  for the long baseline case can also be settled as:

$$\mathbf{x} = (\mathbf{r}_r^T, \mathbf{v}_r^T, Z_r, G_{N,r}, G_{E,r}, Z_b, G_{N,b}, G_{E,b}, \mathbf{I}^T, \mathbf{B}_1^T, \mathbf{B}_2^T, \mathbf{B}_5^T)^T \quad (\text{E.7.25})$$

where  $Z_r$  and  $Z_b$  are ZTD (zenith total delay) at the rover and base-station sites,  $G_{N,r}$ ,  $G_{E,r}$ ,  $G_{N,b}$  and  $G_{E,b}$  are the north and east components of tropospheric gradients.  $\mathbf{I} = (I_{rb}^1, I_{rb}^2, \dots, I_{rb}^m)^T$  is the SD vertical ionospheric delay in  $L_1$  frequency (m) as well.

The measurement model vector  $h(x)$  and the matrix of partial derivatives  $H(x)$  can be written as:

$$\mathbf{h}(x) = (\mathbf{h}_{\phi,1}^T, \mathbf{h}_{\phi,2}^T, \mathbf{h}_{\phi,5}^T, \mathbf{h}_{P,1}^T, \mathbf{h}_{P,2}^T, \mathbf{h}_{P,5}^T)^T \quad (\text{E.7.26})$$

$$\mathbf{h}_{\phi,i} = \begin{pmatrix} \rho_{rb}^{12} + T_{rb}^{12} - \gamma_k(m_l^1 I_{rb}^1 - m_l^2 I_{rb}^2) + \lambda_i(B_{rb,i}^1 - B_{rb,i}^2) + d\phi_{rb,i}^{12} \\ \rho_{rb}^{13} + T_{rb}^{13} - \gamma_k(m_l^1 I_{rb}^1 - m_l^3 I_{rb}^3) + \lambda_i(B_{rb,i}^1 - B_{rb,i}^3) + d\phi_{rb,i}^{13} \\ \vdots \\ \rho_{rb}^{1m} + T_{rb}^{1m} - \gamma_k(m_l^1 I_{rb}^1 - m_l^m I_{rb}^m) + \lambda_i(B_{rb,i}^1 - B_{rb,i}^m) + d\phi_{rb,i}^{1m} \end{pmatrix} \quad (\text{E.7.27})$$

$$\mathbf{h}_{P,i} = \begin{pmatrix} \rho_{rb}^{12} + T_{rb}^{12} + \gamma_k(m_l^1 I_{rb}^1 - m_l^2 I_{rb}^2) \\ \rho_{rb}^{13} + T_{rb}^{13} + \gamma_k(m_l^1 I_{rb}^1 - m_l^3 I_{rb}^3) \\ \vdots \\ \rho_{rb}^{1m} + T_{rb}^{1m} + \gamma_k(m_l^1 I_{rb}^1 - m_l^m I_{rb}^m) \end{pmatrix} \quad (\text{E.7.28})$$

$$\mathbf{H}(x) = \begin{pmatrix} -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{D}\mathbf{M}_{T,r} & \mathbf{D}\mathbf{M}_{T,b} & -\gamma_1\mathbf{D}\mathbf{M}_I & \lambda_1\mathbf{D} \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{D}\mathbf{M}_{T,r} & \mathbf{D}\mathbf{M}_{T,b} & -\gamma_2\mathbf{D}\mathbf{M}_I & \lambda_2\mathbf{D} \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{D}\mathbf{M}_{T,r} & \mathbf{D}\mathbf{M}_{T,b} & -\gamma_5\mathbf{D}\mathbf{M}_I & \lambda_5\mathbf{D} \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{D}\mathbf{M}_{T,r} & \mathbf{D}\mathbf{M}_{T,b} & \gamma_1\mathbf{D}\mathbf{M}_I & \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{D}\mathbf{M}_{T,r} & \mathbf{D}\mathbf{M}_{T,b} & \gamma_2\mathbf{D}\mathbf{M}_I & \\ -\mathbf{D}\mathbf{E} & \mathbf{0} & \mathbf{D}\mathbf{M}_{T,r} & \mathbf{D}\mathbf{M}_{T,b} & \gamma_5\mathbf{D}\mathbf{M}_I & \end{pmatrix} \quad (\text{E.7.29})$$

where:

$$\gamma_k = \lambda_k^2 / \lambda_1^2$$

$$\mathbf{M}_{T,r} = \begin{pmatrix} m_{WG,r}^1(El_r^1) & m_{W,r}^1(El_r^1) \cot E l_r^1 \cos A z_r^1 & m_{W,r}^1(El_r^1) \cot E l_r^1 \sin A z_r^1 \\ m_{WG,r}^2(El_r^2) & m_{W,r}^2(El_r^2) \cot E l_r^2 \cos A z_r^2 & m_{W,r}^2(El_r^2) \cot E l_r^2 \sin A z_r^2 \\ \vdots & \vdots & \vdots \\ m_{WG,r}^m(El_r^m) & m_{W,r}^m(El_r^m) \cot E l_r^m \cos A z_r^m & m_{W,r}^m(El_r^m) \cot E l_r^m \sin A z_r^m \end{pmatrix}$$

$$\mathbf{M}_I = (m_I^1, m_I^2, \dots, m_I^m)^T$$

The time update of EKF for the long-baseline case is expressed as:

$$\mathbf{F}_k^{k+1} = \begin{pmatrix} \mathbf{I}_{3 \times 3} & \mathbf{I}_{3 \times 3} \tau_r & & & \\ & \mathbf{I}_{3 \times 3} & & & \\ & & \mathbf{I}_{6 \times 6} & & \\ & & & \mathbf{I}_{m \times m} & \\ & & & & \mathbf{I}_{(3m-3) \times (3m-3)} \end{pmatrix} \quad (\text{E.7.30})$$

$$\mathbf{Q}_k^{k+1} = \begin{pmatrix} \mathbf{0}_{3 \times 3} & & & \\ & \mathbf{Q}_v & & \\ & & \mathbf{Q}_T & \\ & & & \mathbf{Q}_I \\ & & & & \mathbf{0}_{(3m-3) \times (3m-3)} \end{pmatrix} \quad (\text{E.7.31})$$

where  $\mathbf{Q}_T$  and  $\mathbf{Q}_I$  are the process noise covariance matrixes of the ionosphere and the troposphere terms. In the equation, the ZTD and gradient parameters for the rover and the base-station and SD vertical ionospheric delays for each satellites are simply modeled as random-walk. In addition to estimate ionosphere and troposphere terms, a "Partial fixing" feature was added for long baseline processing in version 2.4.1. It means that only the some partial portion of all ambiguities are resolved into integer values. Other ambiguities except for the fixed are still pending as float values. To determine whether a ambiguity fixed or not fixed, a simple criterion by using the satellite elevation angle is implemented in RTKLIB. If a satellite is under a threshold of the elevation, the ambiguities of the satellite are not fixed. Only the ambiguities of satellites over the threshold are resolved to integer. The elevation threshold for the ambiguity resolution can be set as the processing option "Min Elevation to Fix Amb" as well as "Min Elevation to Hold Amb" to control the "Fix and Hold" feature.

## (7) Moving-baseline model

The moving baseline mode is usually used if both of the rover and the base station receivers are moving and the only relative position of the rover with respect to the base station is required. The moving-

baseline mode can be utilized to determine the precise attitude by mounting two antennas to a moving platform. In RTKLIB, the moving-baseline mode is applied if the processing option "Positioning Mode" is set to "Moving-Base".

In the moving-baseline mode, the base-station position is not fixed but is estimated by the single point positioning process on epoch-by-epoch basis. Once the base station position is obtained, the base station position fixed to the estimate position and the rover position is estimated by short baseline kinematic mode described in (1)-(5). In this case only the relative position is meaningful, that means the absolute position solutions of rover and base-station have only the accuracy as same as the solution by the point pointing mode.

In addition to the simple implementation for the moving-baseline mode, RTKLIB corrects the time difference between the rover and the base station. The rover receiver and the base station receiver are not synchronized. The receiver clock difference usually reaches 2 ms as maximum. The unsynchronized clocks bring the accuracy degradation in case of very fast moving platform. To correct the clock difference, the base station position  $\mathbf{r}_b$  is corrected before the baseline processing by:

$$\mathbf{r}_b(t_r) = \mathbf{r}_b(t_b) + \mathbf{v}_b(t_b)(t_r - t_b) \quad (\text{E.7.32})$$

where  $t_r$  and  $t_b$  are the signal reception time at the rover and the base station estimated by the single point positioning process.  $\mathbf{v}_b(t_b)$  is also the velocity of the base station estimated with Doppler measurements. For the attitude determination by the moving baseline mode, the baseline length constraint can be applied if the processing option "Baseline Length Constraint" enabled. The constraint applies the following pseudo-measurement in the EKF measurement update.

$$\mathbf{y} = (r_{\text{baseline}}) \quad (\text{E.7.33})$$

$$\mathbf{h}(x) = (|\mathbf{r}_r(t_r) - \mathbf{r}_b(t_r)|) \quad (\text{E.7.34})$$

$$\mathbf{H} = \left( \frac{(\mathbf{r}_r(t_r) - \mathbf{r}_b(t_r))^T}{|\mathbf{r}_r(t_r) - \mathbf{r}_b(t_r)|} \right) \quad (\text{E.7.35})$$

$$\mathbf{R} = (\sigma_r^2) \quad (\text{E.7.36})$$

where  $r_{\text{baseline}}$  is the given pre-determined baseline length (m) and  $\sigma_r$  is the constraint of the baseline length (m). To cope with the non-linearity in case of a very short length baseline, iterative measurement update of EKF is supported by setting the processing option "Number of Filter Iteration" to more than 1.



## E.8 PPP (Precise Point Positioning)

In PPP modes, RTKLIB also use EKF estimation process similar to the schemes described in Appendix E.7. The difference from the baseline processing, PPP employs ZD (zero-difference) measurement equations like the single point positioning model instead of DD.

### (1) ZD measurement models for PPP

The ionosphere-free LC phase-range  $\Phi_{r,LC}^s$  and pseudorange  $P_{r,LC}^s$  measurements for the satellite  $s$  are expressed by using (E.3.4) and (E.3.2) as:

$$\begin{aligned}\Phi_{r,LC}^s &= \rho_r^s + c(dt_r(t_r) - dT^s(t^s)) + T_r^s + B_{r,LC}^s + d\Phi_{r,LC}^s + \varepsilon_\phi \\ P_{r,LC}^s &= \rho_r^s + c(dt_r(t_r) - dT^s(t^s)) + T_r^s + \varepsilon_p\end{aligned}\quad (\text{E.8.1})$$

where the ionosphere terms in (E.3.4) and (E.3.2) are eliminated by forming the ionosphere-free LC described in E.5 (7).  $B_{r,LC}^s$  is the carrier-phase bias in m and  $d\Phi_{r,LC}^s$  is the ionosphere-free LC of  $L_i$  and  $L_j$  carrier-phase correction terms expressed as:

$$\begin{aligned}d\Phi_{r,LC}^s = & -\left(C_i \mathbf{d}_{r,pco,i} + C_j \mathbf{d}_{r,pco,j}\right)^T \mathbf{e}_{r,enu}^s + \left(\mathbf{E}^s \left(C_i \mathbf{d}_{pco,i}^s + C_j \mathbf{d}_{pco,j}^s\right)\right)^T \mathbf{e}_r^s + \left(C_i d_{r,pcv,i}(El) + C_j d_{r,pcv,j}(El)\right) + \\ & \left(d_{pcv,i}^s(\theta) + d_{pcv,j}^s(\theta)\right) - \mathbf{d}_{r,disp}^T \mathbf{e}_{r,enu}^s + \left(C_i \lambda_i + C_j \lambda_j\right) \phi_{pw}\end{aligned}\quad (\text{E.8.2})$$

### (2) Receiver antenna phase center model

Figure E.8-1 shows the receiver antenna PCO (phase center offset) and PCV (phase center variation). The antenna PCO is defined as the relative position of the receiver antenna phase center with respect to the ARP (antenna reference point). The PCV is defined as the excess phase delay by the antenna depending on the elevation and azimuth angle. The PCO and PCV values for various antenna types have been measured by the appropriate antenna calibration process and provided by some standard formats. Current version RTKLIB supports the NGS PCV and the ANTEX format for the antenna model including PCO and PCV data. The PCO value is usually provide in the local coordinates at the receiver position. So the  $L_i$  PCO of the receiver antenna in ECEF  $\mathbf{d}_{r,pco,i}$  is expressed as:

$$\mathbf{d}_{r,pco,i} = \mathbf{E}_r^T \mathbf{d}_{r,pco,i,enu} \quad (\text{E.8.3})$$

where  $\mathbf{E}_r$  is the ECEF to local coordinates rotation matrix given by (E.2.10) and  $\mathbf{d}_{r,pco,i,enu}$  is  $L_i$  receiver

antenna PCO expressed in the local coordinates. The  $L_i$  receiver antenna PCV  $d_{r,pcv,i}(El)$  at the elevation angle  $El$  is derived by the linear interpolation of given PCV values depending on the elevation angles as:

$$d_{r,pcv,i}(El) = \frac{(El - El_i)d_{r,pcv,i}(El_i) + (El_{i+1} - El)d_{r,pcv,i}(El_{i+1})}{El_{i+1} - El_i} \quad (\text{E.8.4})$$

where  $El_i \leq El < El_{i+1}$ .

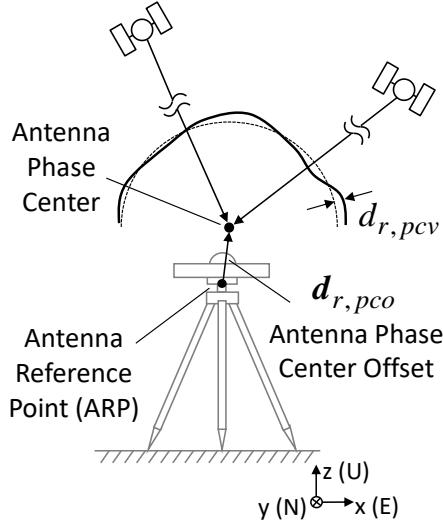


Figure F.8-1 Receiver Antenna Phase Center

### (3) Satellite antenna phase center model

Figure F.8-1 shows the GNSS satellite antenna PCO and PCV, which are also provided as the ANTEX file typically by IGS (international GNSS service) like the receiver antenna. RTKLIB also can import such the antenna model for satellites. The satellite antenna PCO is usually given in satellite fixed coordinates with respect to the satellite CoM (center of mass) shown in Figure F.8-3. So the satellite antenna PCO in the satellite fixed coordinates should be transferred to ECEF coordinates. The PCV is also expressed with respect to the nadir angle.

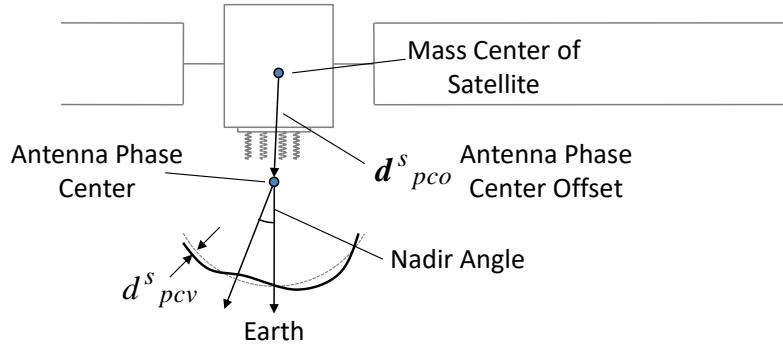


Figure F.8.2 Satellite Antenna Phase Center

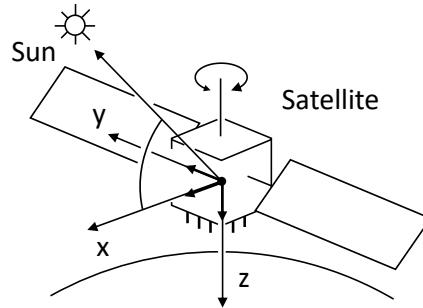


Figure F.8.3 Satellite Body-Fixed Coordinate System

Assuming the nominal satellite attitude control mode, the coordinates transformation matrix  $E_s$  from the satellite body-fixed coordinates to ECEF coordinates is given as:

$$\mathbf{e}_z^s = -\frac{\mathbf{r}^s}{|\mathbf{r}^s|} \quad (E.8.5)$$

$$\mathbf{e}_s = \frac{\mathbf{r}_{sun} - \mathbf{r}^s}{|\mathbf{r}_{sun} - \mathbf{r}^s|} \quad (E.8.6)$$

$$\mathbf{e}_y^s = \frac{\mathbf{e}_z^s \times \mathbf{e}_s}{|\mathbf{e}_z^s \times \mathbf{e}_s|} \quad (E.8.7)$$

$$\mathbf{e}_x^s = \mathbf{e}_y^s \times \mathbf{e}_z^s \quad (E.8.8)$$

$$\mathbf{E}_s = (\mathbf{e}_x^s, \mathbf{e}_y^s, \mathbf{e}_z^s) \quad (E.8.9)$$

where  $\mathbf{r}_{sun}$  is the sun position in ECEF coordinates. The nadir angle  $\theta$  can also be derived as:

$$\theta = \arccos \frac{\mathbf{e}_r^{s^T} \mathbf{r}^s}{|\mathbf{r}^s|} \quad (E.8.10)$$

#### (4) Site displacement by earth tides

The position of a receiver fixed to the ground is varied by the earth tide effects. The tides effects are usually neglected for the baseline analysis because the DD can cancel the almost all of the effects. However, in the PPP mode, the effect should be incorporated in the model because the amplitude sometimes reaches several 10 cm as the vertical component. Figure E.8-2 shows the example of the site displacement effect by earth tides.

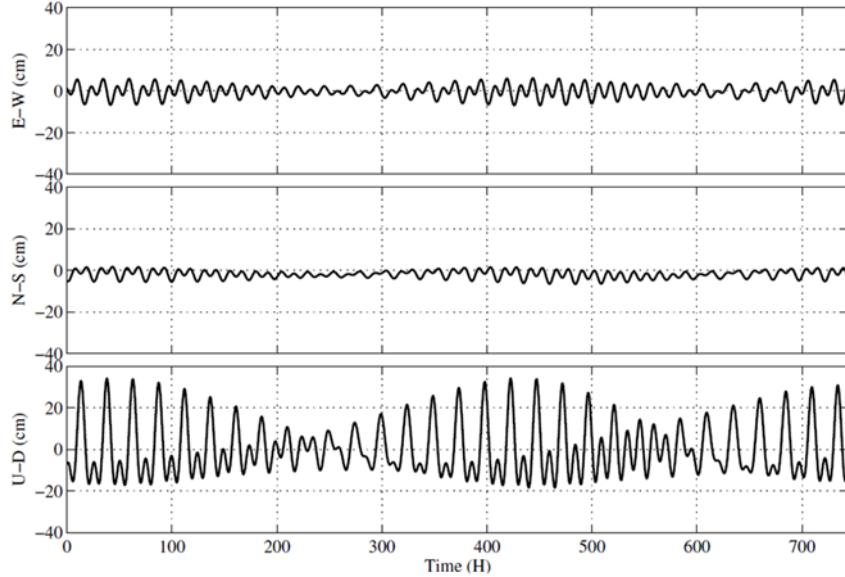


Figure E.8-2 Displacement by Earth Tides

In current version RTKLIB, solid earth tides, OTL (ocean tide loading) and pole tides are modeled and considered if the processing option "Earth Tide Correction" to "Solid" or "Solid/OTL". The earth tide models is based on IERS Conventions 1996 [73]. By setting the compiler option -DIERS\_MODEL and rebuilding APs, RTKLIB uses the IERS Conventions 2010 [74] solid earth tides model by linking the FORTRAN subroutine DEHANTTIDEINEL provided as IERS. For the OTL model, OTL coefficients as provided as the BLQ format [63] should be input as specified as the processing option "OTL BLQ File".

##### (5) Phase windup correction

The phase windup effect is the phase advance and delay by the relative rotation between the receiver and satellite antennas. The phase windup is modeled as:

$$\mathbf{E}_r = (\mathbf{e}_{r,x}^T, \mathbf{e}_{r,y}^T, \mathbf{e}_{r,z}^T)^T \quad (E.8.11)$$

$$\mathbf{E}^s = (\mathbf{e}_x^{sT}, \mathbf{e}_y^{sT}, \mathbf{e}_z^{sT})^T \quad (E.8.12)$$

$$\mathbf{D}^s = \mathbf{e}_x^s - \mathbf{e}_u^s(\mathbf{e}_u^s \cdot \mathbf{e}_x^s) - \mathbf{e}_u^s \times \mathbf{e}_y^s \quad (E.8.13)$$

$$\mathbf{D}_r = \mathbf{e}_{r,x} - \mathbf{e}_r^s(\mathbf{e}_r^s \cdot \mathbf{e}_{r,x}) + \mathbf{e}_r^s \times \mathbf{e}_{r,y} \quad (E.8.14)$$

$$\phi_{pw} = sign(\mathbf{e}_r^s \cdot (\mathbf{D}^s \times \mathbf{D}_r)) \arccos \frac{\mathbf{D}^s \cdot \mathbf{D}_r}{\|\mathbf{D}^s\| \|\mathbf{D}_r\|} / 2\pi + N \quad (E.8.15)$$

where  $N$  is the integer ambiguity, which is determined as to avoiding cycle-jumps.

(6) Estimation of receiver position by PPP

By using EKF, the unknown state vector  $x$  for the PPP case is settled as:

$$\mathbf{x} = (\mathbf{r}_r^T, \mathbf{v}_r^T, cdt_r, Z_r, G_{N,r}, G_{E,r}, \mathbf{B}_{LC}^T)^T \quad (\text{E.8.16})$$

where  $Z_r$  is ZTD (zenith total delay),  $G_{N,r}$  and  $G_{E,r}$  are the north and east components of tropospheric gradients.  $\mathbf{B}_{LC} = (B_{r,LC}^1, B_{r,LC}^2, \dots, B_{r,LC}^m)^T$  is ionosphere-free LC of ZD (zero-differenced) carrier-phase biases (m).

The measurement vector  $y$  is also defined with ZD ionosphere-free LC phase-range and pseudorange measurements with (E.5.21) and (E.5.22) as:

$$\mathbf{y} = (\Phi_{LC}^T, \mathbf{P}_{LC}^T)^T \quad (\text{E.8.17})$$

where:

$$\Phi_{LC} = (\Phi_{r,LC}^1, \Phi_{r,LC}^2, \Phi_{r,LC}^3, \dots, \Phi_{r,LC}^m)^T$$

$$\mathbf{P}_i = (P_{r,LC}^1, P_{r,LC}^2, P_{r,LC}^3, \dots, P_{r,LC}^m)^T$$

The measurement model vector  $h(x)$  and the matrix of partial derivatives  $H(x)$  can be written as:

$$\mathbf{h}(x) = (\mathbf{h}_\phi^T, \mathbf{h}_p^T)^T \quad (\text{E.8.18})$$

$$\mathbf{h}_\phi = \begin{pmatrix} \rho_r^1 + c(dt_r - dT^1) + T_r^1 + B_{r,LC}^1 + d\Phi_{r,LC}^1 \\ \rho_r^2 + c(dt_r - dT^2) + T_r^2 + B_{r,LC}^2 + d\Phi_{r,LC}^2 \\ \vdots \\ \rho_r^m + c(dt_r - dT^m) + T_r^m + B_{r,LC}^m + d\Phi_{r,LC}^m \end{pmatrix} \quad (\text{E.8.19})$$

$$\mathbf{h}_p = \begin{pmatrix} \rho_r^1 + c(dt_r - dT^1) + T_r^1 \\ \rho_r^2 + c(dt_r - dT^2) + T_r^2 \\ \vdots \\ \rho_r^m + c(dt_r - dT^m) + T_r^m \end{pmatrix} \quad (\text{E.8.20})$$

$$\mathbf{H}(x) = \begin{pmatrix} -\mathbf{DE} & \mathbf{0} & \mathbf{1} & \mathbf{DM}_T & \mathbf{I} \\ -\mathbf{DE} & \mathbf{0} & \mathbf{1} & \mathbf{DM}_T & \mathbf{0} \end{pmatrix} \quad (\text{E.8.21})$$

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_{\Phi,LC} & \\ & \mathbf{R}_{P,LC} \end{pmatrix} \quad (\text{E.8.22})$$

where:

$$\mathbf{1} = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$\mathbf{M}_T = \begin{pmatrix} m_{WG,r}^1(El_r^1) & m_{W,r}^1(El_r^1) \cot E l_r^1 \cos A z_r^1 & m_{W,r}^1(El_r^1) \cot E l_r^1 \sin A z_r^1 \\ m_{WG,r}^2(El_r^2) & m_{W,r}^2(El_r^2) \cot E l_r^2 \cos A z_r^2 & m_{W,r}^2(El_r^2) \cot E l_r^2 \sin A z_r^2 \\ \vdots & \vdots & \vdots \\ m_{WG,r}^m(El_r^m) & m_{W,r}^m(El_r^m) \cot E l_r^m \cos A z_r^m & m_{W,r}^m(El_r^m) \cot E l_r^m \sin A z_r^m \end{pmatrix}$$

$$\mathbf{R}_{\phi,LC} = \text{diag}(3\sigma_{\phi,1}^{1,2}, 3\sigma_{\phi,1}^{2,2}, \dots, 3\sigma_{\phi,1}^{m,2})$$

$$\mathbf{R}_{P,LC} = \text{diag}(3\sigma_{P,1}^{1,2}, 3\sigma_{P,1}^{2,2}, \dots, 3\sigma_{P,1}^{m,2})$$

$\sigma_{\phi,1}^s$  : standard deviation of  $L_1$  phase-range measurement error (m)

$\sigma_{P,1}^s$  : standard deviation of  $L_1$  pseudorange measurement error (m)

By using the EKF formulation and the similar time update processes described in E.7, the unknown parameters including the receiver position and velocity, the receiver clock bias, the troposphere parameters and the ionosphere-free LC carrier-phase biases are estimated.

Note that to resolve the integer ambiguity in the carrier-phase biases like the baseline processing cases, the additional information for the satellite-side FCB (fractional cycle bias) is needed. The process is sometimes called as PPP-AR (ambiguity resolution). Current version RTKLIB only supports the wide-lane FCB and the IRC (integer recovery clock) products provided by CNES. The implementation is experimental and unstable. The PPP-AR feature is enabled in case of the processing option "Integer Ambiguity Resolution" is set to "PPP-AR" only in the post-processing mode.

## Appendix F Notes on Config Parameters

I won't try to give a comprehensive explanation of all the input options here, but will explain the ones I have found useful to adjust in my experiments and include a little about why I chose the values I did. I describe them as they appear in the configuration file rather than how they appear in the RTKNAVI GUI menu but the comments apply to both. I created this list by comparing my latest config files to the default config file and noting which settings were different. The values in the list below are the values I use in my config file for a 5 Hz rover measurement rate. The same config files can be used for either RTKNAVI, RTKPOST, or RNX2RTKP.

The settings and options highlighted in blue below are available only in the demo5 code and not in the release code but otherwise much of what I describe below will apply to either code. Most of my work is done for RTK solutions with Ublox M8T and F9P receivers and short baselines and these settings will more directly apply to these combinations but should be useful at least as a starting point for other scenarios.

These notes are intended to be used as a supplement to the earlier descriptions so please refer to them for any of the input parameters not covered here.

### SETTING1:

**pos1-posmode = static, kinematic, static-start, movingbase, fixed**

If the rover is stationary, use "static". If it is moving, use "kinematic" or "static-start". "Static-start" will assume the rover is stationary until first fix is achieved and then switch to dynamic mode, allowing the kalman filter to take advantage of the knowledge that the rover is not moving initially. You can use "movingbase" if the base is moving as well as the rover, but it is not required unless the base is moving long distances. I often find that "kinematic" gives better solutions than "movingbase" even when the base is moving. "Movingbase" mode is not compatible with dynamics, so be sure not to enable both at the same time. If the base and rover remain at a fixed distance apart, set "pos2-baselen" and "pos2-basesig" when in "movingbase" mode. Use "fixed" if you know the rover's exact location and are only interested in analyzing the residuals.

**pos1-frequency = l1, l1+l2, l1+l2+l5**

Use "l1" for single frequency receivers, "l1+l2/E5b" if the rover is using L2 GPS/GLONASS/Beidou and/or Galileo E5b., and "l1+l2+l5" for L5 GPS/GLONASS/BeiDou and or Galileo E5a

**pos1-soltype = forward, backward, combined**

This is the direction in time that the kalman filter is run in post-processing. The “combined” mode first runs the filter forward, then backwards and combines the results. For each epoch, if both directions have a fix, then the combined result is the average of the two with a fixed status unless the difference between the two is too large in which case the status will be float. If only one direction has a fix, that value will be used and the status will be fixed. If both directions are float then the average will be used and the status will be float. Results are not always better with combined because a false fix when running in either direction will usually cause the combined result to be float and incorrect. The primary advantage of combined is that it will usually give you fixed status right to the beginning of the data while the forward only solution will take some time to converge. In “combined” mode the bias states are reset before starting the backwards run to maximize the independence of the forward and backwards solutions. In “combined mode-no phase reset”, the bias states are not reset to avoid needing to re-converge at the start of the backwards solution.. I only use the “backward” setting for debug when I am having trouble getting an initial fix and want to know what the correct satellite phase-biases are.

**pos1-elmask = 15 (degrees)**

Minimum satellite elevation for use in calculating position. I usually set this to 10-15 degrees to reduce the chance of bringing multipath into the solution but this setting will be dependent on the rover environment. The more open the sky view, the lower this value can be set to. Low elevation satellites will also have larger atmospheric errors so this is another reason to exclude the lowest elevation satellites.

**pos1-snrmask-r, pos1-snrmask-b = off,on**

Minimum satellite SNR for rover (-r) and base(-b) for use in calculating position. The optimal value will vary with receiver type and antenna type so I leave it off in the defaults but often enable it to improve results when working with more challenging data sets to remove low quality satellites.

**pos1-snrmask\_L1 =35,35,35,35,35,35,35,35**

Set SNR thresholds for each five degrees of elevation. I usually leave all values the same and pick something between 30 and 38 db depending on what the nominal SNR is. These values are only used if pos1-snrmask\_x is set to on. If you are using dual frequencies, you will need to also set “pos1-snrmask\_L2 and/or pos1-snrmask\_L5”

**pos1-dynamics = on**

Enabling rover dynamics adds velocity and acceleration states to the kalman filter for the rover. It will improve “kinematic” and “static-start” results, but will have no effect on “static” mode. Be sure to set

“prnacelh” and “prnaccelv” appropriately for your rover acceleration characteristics. Rover dynamics is not compatible with “movingbase” mode, so turn it off when using that mode.

#### **pos1-posopt1 = off, on (Sat PCV)**

Set whether the satellite antenna phase center variations are used or not. Leave this off for RTK since the antenna offsets will cancel. Recommended for PPP, but if set to on, be sure to specify the satellite antenna PCV file in the files parameters.

#### **pos1-posopt2 = off, on (Rec PCV)**

Set whether the receiver antenna phase center variations are used or not. If set to on, you need to specify the receiver antenna PCV file in the files parameters and the type of receiver antenna for base and rover in the antenna section. Only survey grade antennas are included in the antenna file available from IGS so only use this if your antenna is in the file. It primarily affects accuracy in the z-axis so it can be important if you care about height. You can leave this off if both antennas are the same since they will cancel.

#### **pos1-posopt5 = off, on (RAIM FDE)**

If the residuals for any satellite exceed a threshold, that satellite is excluded. This will only exclude satellites with very large errors but requires a fair bit of computation so I usually leave this disabled.

#### **pos1-exclsats=**

If you know a satellite is bad you can exclude it from the solution by listing it here. I only use this in rare cases for debugging if I suspect a satellite is bad. You can also force an unhealthy satellite to be used by prefixing the satellite number with “+”.

#### **pos1-navsys = 7, 15,**

I almost always include all satellite systems available since more information is generally better. The exception is that SBAS should not be enabled if using the EGNOS satellites.

#### **SETTING2:**

##### **pos2-armode = continuous, fix-and-hold**

Integer ambiguity resolution method. “Continuous” mode does not take advantage of fixes to adjust the phase bias states so it is the most immune to false fixes. “Fix-and-hold” does use feedback from the fixes to help track the ambiguities. I prefer to use “fix-and-hold” for dynamic rovers and adjust the tracking gain (pos2-varholdamb) low enough to minimize the chance of a false fix. If “armode” is not set to “fix-and-hold” then any of the options below that refer to holds don’t apply.

##### **pos2-varholdamb=0.1, 1.0 (meters)**

In the demo5 code, the tracking gain for fix-and-hold can be adjusted with this parameter. It is actually a variance rather than a gain, so larger values will give lower gain. 0.1 is the default value, anything over 100 will have very little effect. This value is used as the variance for the pseudo-measurements generated during a hold which provide feedback to drive the bias states in the kalman filter towards integer values. I find that values from 0.1 to 1.0 provides enough gain to assist with tracking while still avoiding tracking of false fixes in most cases.

**pos2-gloarmode = on, fix-and-hold, autocal**

Integer ambiguity resolution for the GLONASS sats. If your receiver types are identical or both have zero Glonass hardware biases, you can set this to "on". If your receiver biases are different you will need to account for the inter-channel biases. The easiest way do do this is to set this parameter to "fix-and-hold" In this case the GLONASS sats will not be used for ambiguity resolution until after the inter-channel biases have been calibrated which begins after the first hold. Alternatively, you can set this parameter to "autocal" and then specify the differential hardware offset between base and rover with the "pos2-arthres2" parameter. This will allow the GLONASS sats to be used for ambiguity resolution immediately and hence will generally give better performance than the "fix-and-hold" setting. The "autocal" feature can also be used to determine the inter-channel biases with a zero or short baseline using an iterative approach.

**pos2-gainholdamb=0.01**

In the demo5 code, the gain of the inter-channel bias calibration for the GLONASS satellites can be adjusted with this parameter.

**pos2-arthres = 3**

This is the threshold used to determine if there is enough confidence in the ambiguity resolution solution to declare a fix. It is the ratio of the squared residuals of the second-best solution to the best solution. I generally leave the nomiinal ratio at the default value of 3.0 and adjust all the other parameters to work around this one. Although a larger AR ratio indicates higher confidence than a low AR ratio, there is not a fixed relationship between the two. The larger the errors in the kalman filter states, the lower the confidence in that solution will be for a given AR ratio. Generally the errors in the kalman filter will be largest when it is first converging so this is the most likely time to get a false fix. Reducing pos2-arthres1 can help avoid this.

**pos2-arthresmin, pos2-arthresmax=1.5, 10**

If these values are set equal to pos2-arthres then the ambiguity resolution threshold will be fixed. Otherwise the threshold will be adjusted based on the number of satellites used in the ambiguity resolution. The nominal value is used for 8 satellite pairs and decreases with more satellites and increases with less satellite.

The adjusted value is clipped by the min and max threshold limits. The adjustment rates are based on those used by the FFRT method but are adjusted only for the number of satellites and not for model strength.

#### **pos2-arfilter = on**

Setting this to on will qualify new sats or sats recovering from a cycle-slip. If a sat significantly degrades the AR ratio when it is first added, its use for ambiguity resolution will be delayed. Turning this on should allow you to reduce “arlockcnt” which serves a similar purpose but with a blind delay count.

#### **pos2-arthres1 = 0.004-0.10**

Integer ambiguity resolution is delayed until the variance of the position state has reached this threshold. It is intended to avoid false fixes before the bias states in the kalman filter have had time to converge. It is particularly important to set this to a relatively low value if you have set eratio1 to values larger than 100 and are using a single constellation or single frequency solution. If you see AR ratios of zero extending too far into your solution, you may need to increase this value since it means ambiguity resolution has been disabled because the threshold has not been met yet. I find 0.004 to 0.10 usually works well for me but if your measurements are lower quality you may need to increase this to avoid overly delaying first fix or losing fix after multiple cycle slips have occurred.

#### **pos2-arthres2**

Relative GLONASS hardware bias in meters per frequency slot. This parameter is only used when pos2-gloarmode is set to “autocal” and is used to specify the inter-channel bias between two different receiver manufacturers. To find the appropriate values for common receiver types, as well as how to use this parameter for an iterative search to find values for receiver types not specified, see [this post](#).

#### **pos2-arthres3 = 1e-9,1e-7**

Initial variance of the GLONASS hardware bias state. This parameter is only used when pos2-gloarmode is set to “autocal”. A smaller value will give more weight to the initial value specified in pos2-arthres2. I use 1e-9 when pos2-arthres2 is set to a known bias, and 1e-7 for iterative searches.

#### **pos2-arthres4 = 0.00001,0.001**

Kalman filter process noise for the GLONASS hardware bias state. A smaller value will give more weight to the initial value specified in pos2-arthres2. I use 0.00001 when pos2-arthres2 is set to a known bias, and 0.001 for iterative searches.

#### **pos2-arlockcnt = 0, 5**

Number of samples to delay a new sat or sat recovering from a cycle-slip before using it for integer ambiguity resolution. Avoids corruption of the AR ratio from including a sat that hasn’t had time to converge

yet. Use in conjunction with “arfilter”. Note that the units are in samples, not units of time, so it may need to be adjusted if you change the rover measurement sample rate. I usually set this to zero for u-blox receivers which are very good at flagging questionable observations but set it higher for other receivers.

**pos2-minfixsats = 4**

Minimum number of sats necessary to get a fix. Used to avoid false fixes from a very small number of satellites, especially during periods of frequent cycle-slips.

**pos2-minholdsats = 5**

Minimum number of sats necessary to hold an integer ambiguity result. Used to avoid false holds from a very small number of satellites, especially during periods of frequent cycle-slips.

**pos2-mindropsats = 10**

Minimum number of sats necessary to enable exclusion of a single satellite from ambiguity resolution each epoch. In each epoch a different satellite is excluded. If excluding the satellite results in a significant improvement in the AR ratio, then that satellite is removed from the list of satellites used for AR.

**pos2-rcvstds = on,off**

Experimental feature. Enabling this feature causes the measurement variances for the raw pseudorange and phase measurement observations to be adjusted based on the standard deviation of the measurements as reported by the receiver. This feature is currently only supported for u-blox receivers. The adjustment in variance is in addition to adjustments made for satellite elevation based on the stats-errphaseel parameter.

I generally get better results with this turned off.

**pos2-arelmask = 15**

Functionally no different from the default of zero, since elevations less than “elmask” will not be used for ambiguity resolution but I changed it to avoid confusion.

**pos2-arminfix = 20-100 (5-20\*sample rate)**

Number of consecutive fix samples needed to hold the ambiguities. Increasing this is probably the most effective way to reduce false holds, but will also increase time to first hold and time to reacquire a hold. As the ambiguity tracking gain is reduced (i.e. as pos2-varholdamb is increased), and the number of observations increases, arminfix can be reduced. Note that this value may need to be adjusted if the rover measurement sample rate changes.

**pos2-elmaskhold = 15**

Functionally no different from the default of zero, since elevations less than “elmask” will not be used for holding ambiguity resolution results but I changed it to avoid confusion.

**pos2-aroutcnt = 100 (20\*sample rate)**

Number of consecutive missing samples that will cause the ambiguities to be reset. Again, this value needs to be adjusted if the rover measurement sample rate changes.

**pos2-maxage = 100**

Maximum delay between rover measurement and base measurement (age of differential) in seconds. This usually occurs because of missing measurements from a misbehaving radio link. I've increased it from the default because I found I was often still getting good results even when this value got fairly large, assuming the dropout occurred after first fix-and-hold.

**pos2-rejionno = 1.0-2.0**

Reject a measurement if the kalman filter residual are greater than this value in meters. Previous to the demo5 b33 code, this value was applied without adjustment to both code and phase measurements. In the newer versions, this value is still applied without adjustment to the phase measurements but is multiplied by eratio for the code measurements. This allows it to be set to values appropriate for the phase measurements. I usually set it to 1.0 which is very helpful to catch and reject unflagged cycle slips but occasionally find I need to set it higher. Setting it too low can cause the kalman filter to diverge with low quality data so I have set the default to 2.0 even though I usually use 1.0.

**OUTPUT:****out-solformat = enu, llh, xyz**

I am usually interested in relative distances between rover and base, so set this to "enu". If you are interested in absolute locations, set this to "llh" but make sure you set the exact base location in the "ant2" settings. Be careful with this setting if you need accurate z-axis measurements. Only the llh format will give you a constant z-height if the rover is at constant altitude. "Enu" and "xyz" are cartesian coordinates and so the z-axis follows a flat plane, not the curvature of the earth. This can lead to particularly large errors if the base station is located farther from the rover since the curvature will increase with distance.

**out-outhead = on**

No functional difference to the solution, just output more info to the result file.

**out-outopt = on**

No functional difference to the solution, just output more info to the result file.

**out-outstat = residual**

## RTKLIB Demo5 Manual

No functional difference to the solution, just output residuals to a file. The residuals can be very useful for debugging problems with a solution and can be plotted with RTKPLOT as long as the residual file is in the same folder as the solution file.

**stats-eratio1** = 300  
**stats-eratio2** = 300  
**stats-eratio5** = 300

Ratio of the standard deviations of the pseudorange measurements to the carrier-phase measurements. I have found a larger value works better for low-cost receivers, but that the default value of 100 often work better for more expensive receivers since they have less noisy pseudorange measurements. Larger values tend to cause the kalman filter to converge faster and leads to faster first fixes but it also increases the chance of a false fix. If you increase this value, you should set pos2-arthres1 low enough to prevent finding fixes before the kalman filter has had time to converge. I believe increasing this value has a similar effect to increasing the time constant on a pseudorange smoothing algorithm in that it filters out more of the higher frequencies in the pseudorange measurements while maintaining the low frequency components.

**stats-prnaccelh** = 3.0

If receiver dynamics are enabled, use this value to set the standard deviation of the rover receiver acceleration in the horizontal components. This value should include accelerations at all frequencies, not just low frequencies. It should characterize any movements of the rover antenna, not just movements of the complete rover so it may be larger than you think. It will include accelerations from vibration, bumps in the road, etc as well as the more obvious rigid-body accelerations of the whole rover. It can be estimated by running a solution with this value set to a large value, then examining the accel values in the solution file with RTKPLOT

**stats-prnaccelv** = 1.0

The comments about horizontal accelerations apply even more to the vertical acceleration component since in many applications the intentional accelerations will all be in the horizontal components. It is best to derive this value from actual GPS measurement data rather than expectations of the rigid-body rover. It is better to over-estimate these values than to under-estimate them.

**ant2-postype** = rinexhead, llh, single

This is the location of the base station antenna. If you are only interested in relative distance between base and rover this value does not need to be particularly accurate. For post-processing I usually use the approximate base station location from the RINEX file header. Although labeled approximate, this will

normally be precise if you are using a rinex file from a CORS reference station. Otherwise, if I want absolute position, I first process the base station data against a nearby reference station to get the exact location, then use the "llh" or "xyz" option to specify that location. For real-time processing where I don't know the exact base location, I use the "single" option which uses the single solution from the data to get a rough estimate of base station location.

**ant2-maxaveep = 1**

Specifies the number of samples averaged to determine base station location if "postype" is set to "single". I set this to one to prevent the base station position from varying after the kalman filter has started to converge since that seems to cause long times to first fix. In most cases for post-processing, the base station location will come from the RINEX file header and so you will not use this setting. However if you are working with RTCM files you may need this even for post-processing.

**MISC:****misc-timeinterp =off,on**

Interpolates the base station observations. I generally set this to "on" if the base station observations sample time is larger than 5 seconds.

## Appendix G GNSS Signal Specifications

System	Freq. (MHz)	Signal	Band Width (MHz)	I/Q	Min. Power (dBW)	Modulation	Spreading Code				Navigation Data				Notes	
							Primary (chips)	Second (chips)	Rate (Mcps)	Period	ENC	Data	Rate (sps)	Rate (bps)	FEC	
GPS <sup>[1][2][3]</sup>	1575.42	L1C/A	2.046	Q	-158.5	BPSK (1)	1,023	-	1.023	1ms	-	NAV	50	50	-	
		L1P(Y)	20.46	I	-161.5	BPSK (10)	1week	-	10.23	1week	(Y)	NAV	50	50	-	
		L1M	?	?	?	BOC (10,5)	?	?	5.115	?	Y	?	?	?	?	Block IIR-M-
		L1C-D	30.69	I	-163	BOC (1,1)	10,230	-	1.023	10ms	-	CNAV-2	100	50	1/2	Block III-
		L1C-P	30.69	I	-158.25	TMBOC (6,1,1/11)	10,230	1,800	1.023	18s	-	-	-	-	-	Block III-
GPS <sup>[1][2][3]</sup>	1227.6	L2C/A	2.046	Q	-160.0	BPSK (1)	1,023	-	1.023	1ms	-	(NAV)	-50	-50	-	Block IIR-M-
		L2P(Y)	20.46	I	-164.5/ -161.5	BPSK (10)	1week	-	10.23	1week	(Y)	(NAV)	-50	-50	-	
		L2M	?	?	?	BOC (10,5)	?	?	5.115	?	Y	?	?	?	?	Block IIR-M-
		L2C	2.046	Q/I	-160.0	BPSK (1)	10,230	-	0.5115	20ms	-	(CNAV)	-50	-25	1/2	time mux, Block IIR-M-
		L5-I	20.46	I	-157.9/ -157.0	BPSK (10)	10,230	10	10.23	10ms	-	CNAV	100	50	1/2	Block IIF-
GLONASS <sup>[4]</sup>	1176.45	L5-Q	20.46	Q	-157.9/ -157.0	BPSK (10)	10,230	20	10.23	20ms	-	-	-	-	-	Block IIF-
		L1C/A	1.022	I	-161.0	BPSK	511	-	0.511	1ms	-	NAV	50	50	-	
		L1P	10.22	Q	?	BPSK	5,110,000	-	5.11	1s	(Y)	NAV	50	50	-	
		L2C/A	1.022	I	-167.0	BPSK	511	-	0.511	1ms	-	NAV	50	50	-	
		L2P	10.22	Q	?	BPSK	5,110,000	-	5.11	1s	(Y)	NAV	50	50	-	
Galileo <sup>[5]</sup>	1575.42	L3-I	20.46	I	?	BPSK (10)	10,230	5	10.23	5ms	-	NAV	100	50	1/2	GLONASS-K-
		L3-Q	20.46	Q	?	BPSK (10)	10,230	10	10.23	10ms	-	-	-	-	-	
		E1-A	35.805	Q	?	BOC (15,2.5)	?	?	2.5575	?	Y	G/NAV	?	?	?	PRS
		E1-B	24.552	I	-157.0	CBOC (6,1,1/11)	4,092	-	1.023	4ms	-	I/NAV	250	125	1/2	OS, SoL, CS
		E1-C	24.552	Q	-157.0	CBOC (6,1,1/11)	4,092	25	1.023	100ms	-	-	-	-	-	

System	Freq. (MHz)	Signal	Band Width (MHz)	I/Q	Min. Power (dBW)	Modulation	Spreading Code				Navigation Data				Notes	
							Primary (chips)	Second (chips)	Rate (Mcps)	Period	ENC	Data	Rate (sps)	Rate (bps)	FEC	
Galileo	1191.795	E5a+ E5b	51.15	I	-155.0	8-PSK (10)	10,230	100	10.23	100ms	-	-	-	-	-	AltBOC
	1176.45	E5a-I	20.46	I	-155.0	BPSK (10)	10,230	20	10.23	20ms	-	F/NAV	50	25	1/2	OS, CS
		E5a-Q	20.46	Q		BPSK (10)	10,230	100	10.23	100ms	-	-	-	-	-	
	1207.14	E5b-I	20.46	I	-155.0	BPSK (10)	10,230	4	10.23	4ms	-	I/NAV	250	125	1/2	OS, SoL, CS
		E5b-Q	20.46	Q		BPSK (10)	10,230	100	10.23	100ms	-	-	-	-	-	
		E6-A	?	Q	?	BOC (10,5)	?	?	5.115	?	Y	G/NAV	?	?	?	PRS
	1278.75	E6-B	40.92	I	-155.0	BPSK (5)	5,115	-	5.115	1ms	Y	C/NAV	1,000	?	?	CS
		E6-C	40.92	I		BPSK (5)	5,115	100	5.115	100ms	Y	-	-	-	-	
		L1C/A	2.046	Q	-158.5	BPSK (1)	1,023	-	1.023	1ms	-	NAV	50	50	-	
	1575.42	L1C-D	4.096	I	-163.0	BOC (1,1)	10,230	-	1.023	10ms	-	CNAV-2	100	50	1/2	
QZSS <sup>[6]</sup>		L1C-P	4.096	Q	-158.25	BOC (1,1)	10.23	1,800	1.023	18s	-	-	-	-	-	
		L1-SAIF	2.046	I	-161.0	BPSK (1)	1,023	-	1.023	1ms	-	L1-SAIF	500	250	1/2	
	1227.6	L2C	2.046	I	-160.0	BPSK (1)	10,230	-	0.5115	20ms	-	CNAV	50	50	1/2	chip-by-chip time mux
	1176.45	L5-I	20.46	I	-157.9	BPSK (10)	10,230	10	10.23	10ms	-	CNAV	50	25	1/2	
		L5-Q	20.46	Q	-157.9	BPSK (10)	10,230	20	10.23	20ms	-	-	-	-	-	
BeiDou <sup>[7]</sup>	1278.75	LEX	42.0	I	-155.7	BPSK (5)	10,230	-	2.5575	4ms	-	LEX	2,000	1,744	RS	chip-by-chip time mux
	1561.098	B1	4.092	Q	-163.0	QPSK (2)	2,046	20	2.046	20ms	-	D1-NAV	50	50	BCH	
				I			2,046	-	2.046	1ms	-	D2-NAV	500	500	BCH	GEO
	1207.14	B2	24.0	Q	?	QPSK (10)	2,046	20	2.046	20ms	?	?	?	?	?	
				I			10,230	-	10.23	>0.16s	-	-	-	-	-	
SBAS <sup>[8]</sup>	1268.52	B3	24.0	Q	?	QPSK (10)	10,230	20	10.23	20ms	?	?	?	?	?	
				I			10,230	-	10.23	>0.16s	-	-	-	-	-	
	1575.42	L1	2.046	I	-161.0	BPSK (1)	1,023	-	1.023	1ms	-	SBAS	500	250	1/2	
	1176.45	L5	20.46	I	?	BPSK (10)	10,230	-	10.23	10ms	-	SBAS	500	250	1/2	WAAS

K= -7,...,+6, ENC: encryption, FEC: forward error correction, 1/2: 1/2 convolutional code, RS: Reed Solomon code, BCH: BCH code and interleave

## References

- [1] IS-GPS-200F, Navstar GPS Space Segment/Navigation User Interfaces, September 21, 2011
- [2] IS-GPS-750A, Navstar GPS Space Segment/User Segment L5 Interfaces, June 8, 2010
- [3] IS-GPS-800A, Navstar GPS Space Segment/User Segment L1C Interfaces, June 8, 2010
- [4] Global Navigation Satellite System GLONASS, Interface Control Document Navigational radiosignal In bands L1, L2, (Edition 5.1), 2008
- [5] European GNSS (Galileo) Open Service Signal In Space Interface Control Document, Issue 1, February, 2010
- [6] Quasi-Zenith Satellite System Navigation Service Interface Control Specification for QZSS (IS-QZSS) V1.4, Japan Aerospace Exploration Agency, February, 2012
- [7] BeiDou navigation satellite system signal in space interface control document open service signal B1I (version 1.0), China Satellite Navigation office, December 2012
- [8] RTCA/DO-229C, Minimum operational performance standards for global positioning system/wide area augmentation system airborne equipment, RTCA inc, November 28, 2001
- [9] W. Gurtner, RINEX The Receiver Independent Exchange Format Version 2.10, December 10, 2007
- [10] W. Gurtner and L. Estey, RINEX The Receiver Independent Exchange Format Version 2.11, December 10, 2007
- [11] W. Gurtner and L. Estey, RINEX The Receiver Independent Exchange Format Version 2.12, June 23, 2009
- [12] W. Gurtner and L. Estey, RINEX The Receiver Independent Exchange Format Version 3.00, November 28, 2007
- [13] W. Gurtner and L. Estey, RINEX The Receiver Independent Exchange Format Version 3.01, June 22, 2009
- [14] RINEX The Receiver Independent Exchange Format Version 3.02, International GNSS Service (IGS), RINEX Working Group and Radio Technical Commission for Maritime Services Special Committee 104 (RTCM-SC104), December 10, 2012
- [15] J. Ray and W. Gurtner, RINEX extensions to handle clock information version 3.02, September 2, 2010
- [16] RTCM Recommended Standards for Differential GNSS (Global Navigation Satellite Systems) Service version 2.3, August 20, 2001
- [17] RTCM Standard 10403.1 - Amendment 5, Differential GNSS (Global Navigation Satellite Systems) Services - version 3, July 1, 2011
- [18] RTCM Standard 10403.2, Differential GNSS (Global Navigation Satellite Systems) Services - version 3, February 1, 2013
- [19] UNAVCO, BINEX: Binary exchange format (<http://binex.unavco.org/binex.html>)

- [20] RTCM Recommended Standards for Networked Transport of RTCM via Internet Protocol (Ntrip), version 1.0, September 30, 2004
- [21] NMEA0183, Standard for Interfacing Marine Electronic Devices, version 4.10, June, 2012
- [22] S. Hilla, The extended standard product 3 orbit format (SP3-c), August 17, 2010
- [23] M. Rothacher, R. Schmid, ANTEX: The Antenna Exchange Format Version 1.4, September 15, 2010
- [24] S. Scheer, W. Gurtner and J. Feltens, IONEX: The IONosphere Map EXchange Format Version 1, February 25, 1998
- [25] National Geodetic Survey- Antenna calibrations (<http://www.ngs.noaa.gov/ANTCAL>)
- [26] ESA, EGNOS Message Server (EMS) User Interface Document, November 4, 2005
- [27] NovAtel (<http://www.novatel.com>)
- [28] Hemisphere GPS (<http://www.hemisphergps.com>)
- [29] u-blox (<http://www.u-blox.com>)
- [30] SkyTraq (<http://www.skytraq.com.tw>)
- [31] JAVAD GNSS (<http://www.javad.com>)
- [32] Furuno (<http://www.furunocom>)
- [33] NVS Technologies AG (<http://www.nvs-gnss.com>)
- [34] Google Earth (<http://www.google.com/earth>)
- [35] Microsoft Windows (<http://windows.microsoft.com>)
- [36] LAPACK - Linear Algebra PACKage (<http://www.netlib.org/lapack>)
- [37] Intel (R) Math Kernel Library (Intel (R) MKL) (<http://software.intel.com/en-us/intel-mkl>)
- [38] Embarcadero Technologies (<http://www.embarcadero.com>)
- [39] Ubuntu (<http://www.ubuntu.com>)
- [40] The BSD 2-Clause License (<http://opensource.org/licenses/BSD-2-Clause>)
- [41] NovAtel, OM-20000094 Rev6 OEMV Family Firmware Reference Manual, 2008
- [42] NovAtel, OM-20000129 Rev2 OEM6 Family Firmware Reference Manual, 2011
- [43] NovAtel, OM-20000053 Rev2 MiLLennium GPS Card Software Versions 4.503 and 4.52 Command Descriptions Manual, 2001
- [44] ublox-AG, GPS.G3-X-03002-D, ANTARIS Positioning Engine NMEA and UBX Protocol Specification, Version 5.00, 2003
- [45] NovAtel, OM-20000086 Superstar II Firmware Reference Manual, 2005
- [46] Hemisphere GPS, Crescent Integrator's Manual, December, 2005
- [47] Hemisphere GPS, GPS Technical Reference, Part No. 875-0175-000, Rev.D1, 2008
- [48] SkyTraq, Application Note AN0023 Binary Message of SkyTraq Venus 6 GPS Receiver, ver 1.4.8, August 21, 2008
- [49] SkyTraq, Application Note AN0024 Raw Measurement Binary Message Extension of SkyTraq Venus 6 GPS Receiver, ver. 0.5, October 9, 2009

- [50] Furuno, SBAS/GPS receiver type GW-10 III manual, July 2004
- [51] Javad GNSS, GREIS GNSS Receiver External Interface Specification, Reflects Firmware Version 3.2.0, July 22, 2010
- [52] Javad navigation systems, GPS Receiver Interface Language (GRIL) Reference Guide Rev 2.2, Reflects Firmware Version 2.6.0
- [53] Javad GNSS, User visible changes in the firmware version 3.4.0 since version 3.3.x (NEWS\_3\_4\_0.txt)
- [54] Javad GNSS, GREIS GNSS Receiver External Interface Specification, Reflects Firmware Version 3.4.6, October 9, 2012
- [55] Description of BINR messages which is used by RC program for RINEX files accumulation, NVS
- [56] NVS Technologies AG, GLONASS/GPS/Galileo/Compass/SBAS NV08C receiver series BINR Interface Protocol Specification, ver. 1.3, August, 2012
- [57] The GZIP home page (<http://www.gzip.org>)
- [58] Download page for the RNXCMP software (<http://terras.gsi.go.jp/ja/crx2rnx.html>)
- [59] GNU General Public License version 3, June 29, 2007 (<http://www.gnu.org/licenses/gpl-3.0.en.html>)
- [60] Japan Aerospace Exploration Agency, RINEX ver2-based QZSS Extension (Version 1.00), December 17, 2010
- [61] Japan Aerospace Exploration Agency, RINEX ver3-base QZSS Extension (Version 1.00)
- [62] [IGSMAIL-1943] AC Coordinator, New IGS ERP Format (version 2), July 10, 1998
- [63] Ocean Tide Loading Provider (<http://holt.oso.chalmers.se/loading>)
- [64] P. J. G. Teunissen, A. Kleusberg (Eds.), GPS for Geodesy, 2nd edition, Springer, 1998
- [65] A. Gelb (eds.), Applied Optimal Estimation, The M. I. T. Press, 1974
- [66] P. J. G. Teunissen, The least-square ambiguity decorrelation adjustment: a method for fast GPS ambiguity estimation, J. Geodesy, vol.70, 1995
- [67] X.-W. Chang, X. Yang, and T. Zhou, MLAMBDA: A modified LAMBDA method for integer least-squares estimation, J. Geodesy, vol.79, 2005
- [68] J. Kouba, A guide to using International GNSS Service (IGS) products, May 2009
- [69] D. W. Marquardt, An Algorithm for Least-Square Estimation for Nonlinear Parameters, Journal of the Society for Industrial and Applied Mathematics, 1963
- [70] A. E. Niell, Global mapping functions for the atmosphere delay at radio wavelengths, Journal of geophysical research, 1996
- [71] J. Boehm, A. Niell, P. Tregoning and H. Shuh, Global Mapping Function (GMF): A new empirical mapping function base on numerical weather model data, Geophysical Research Letters, 33, L07304, 2006
- [72] BIPM Bureau International des Poid et Mesures (<http://www.bipm.org>)
- [73] D. D. McCarthy, IERS Technical Note 21, IERS Conventions 1996, July 1996
- [74] G.Petit and B.Luzum (eds.), IERS Technical Note No.36, IERS Conventions (2010), 2010