

# RTKLIB ver. 2.4.0 Manual



Rev. A 2010-08-12

## Contents

1	Overview .....	1
2	System Requirement .....	3
3	Instructions .....	4
3.1	Installation.....	4
3.2	Real-time Positioning with RTKNAVI.....	5
3.3	Configure Input, Output and Log Streams for RTKNAVI.....	17
3.4	Post-Processing Analysis with RTKPOST .....	23
3.5	Configure Positioning Options for RTKNAVI and RTKPOST .....	27
3.6	Convert Receiver Raw Data to RINEX with RTKCONV .....	39
3.7	View and Plot Solutions and Observation Data with RTKPLOT.....	42
3.8	NTRIP Source Table Browser.....	53
3.9	Use Console APs of RTKLIB.....	55
4	Develop and Link user AP with RTKLIB .....	56
	Appendix A Console AP Commands .....	57
A.1	RTKRCV .....	57
A.2	RNX2RTKP.....	60
A.3	POS2KML.....	62
A.4	CONVBIN .....	63
A.5	STR2STR.....	65
	Appendix B File Formats.....	66
B.1	Positioning Solution File.....	66
B.2	SBAS Log File.....	69
B.3	Solution Status File.....	70
B.4	Configuration File .....	72
	Appendix C Library APIs (Application Program Interfaces).....	75

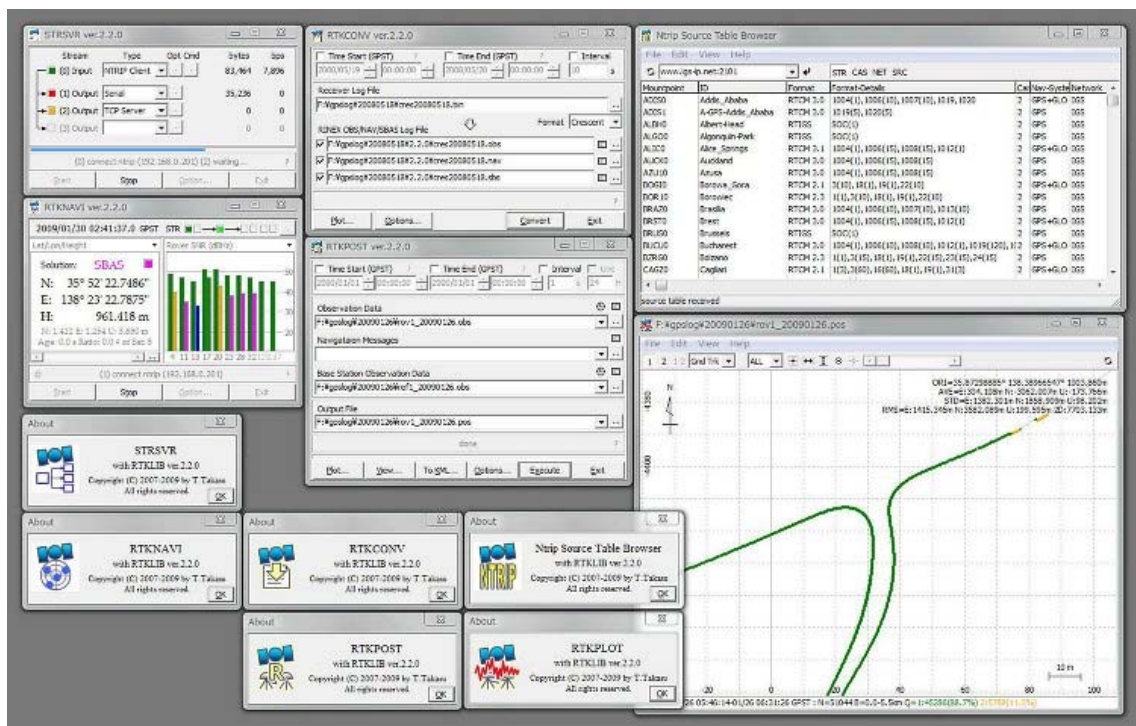
# 1 Overview

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

- (1) Supports standard and precise positioning algorithms with:  
[GPS](#), [GLONASS](#) and [SBAS](#) ([Galileo](#) and [QZSS](#) are supported but disabled in current versions)
- (2) Supports various positioning modes with GNSS for both real-time and post-processing:  
 Single-point, DGPS/DGNSS, Kinematic, Static, Moving-baseline, Fixed, PPP-Kinematic and PPP-Static
- (3) Supports many standard formats and protocols for GNSS:  
[RINEX 2.10, 2.11, 2.12 OBS/NAV/GNAV/HNAV](#), [RINEX 3.00 OBS/NAV](#), [RINEX 3.00 CLK](#), [RTCM v.2.3](#), [RTCM v.3.1](#), [NTRIP 1.0](#), [NMEA 0183](#), [SP3-c](#), [ANTEX 1.3](#), [NGS PCV](#) and [EMS 2.0](#) (see [Release Notes](#) for supported RINEX files and RTCM messages)
- (4) Supports several GNSS receivers' proprietary messages:  
[NovAtel: OEM4/V, OEM3, OEMStar, Superstar II](#), [Hemisphere: Eclipse, Crescent](#), [u-blox: LEA-4T, LEA-5T](#) and [SkyTraq: S1315F](#) (see [Release Notes](#) for supported messages)
- (5) Supports external communication via:  
 Serial, TCP/IP, NTRIP, local log file (record and playback) and FTP/HTTP (automatic download)
- (6) Provides many library functions and APIs:  
 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, Google Earth 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
- (7) Provides GUI and console (command line) APs:
  - RTKNAVI, RTKRCV: real-time positioning
  - RTKPOST, RNX2RTKP: post-processing analysis
  - RTKPLOT: visualization of solutions and observation data
  - RTKCONV, CONVBIN: RINEX translator for RTCM and receiver raw data log
  - STRSVR, STR2STR: communication utility
  - NTRIPSRCBROWS: NTRIP source table browser

- Other positioning utilities

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.



## 2 System Requirement

All of the library functions and APIs are written in ANSI C (C89). The library internally uses standard socket and pthread for Linux/UNIX or winsock and WIN32 thread for Windows. By setting compiler option -DLAPACK or -DMKL, the library uses LAPACK/BLAS or Intel MKL for fast matrix computation. The console APIs are also written in standard C. The library and console APIs can be built on many environments like gcc on Linux. The GUI APIs are written in C++ and utilize Borland VCL for the GUI toolkit. All of the executable binary APIs in the package were built by free edition Borland Turbo C++ 2006 (<http://www.turboexplorer.com>) \* on Windows. The binary APIs were tested on Windows XP SP3 32bit and Windows 7 64bit. The console APIs are also built and tested on Ubuntu 9.04 Linux.

\* Free edition Turbo C++ 2006 is no longer available. We plan to change the GUI toolkit for RTKLIB GUI APIs to open source one in future versions.

## 3 Instructions

### 3.1 Installation

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

<code>rtklib_&lt;ver&gt;</code>	
<code>\src</code>	: Source programs of RTKLIB libraries *
<code>\rcv</code>	: Source programs depending on GPS/GNSS receivers *
<code>\bin</code>	: Executable binary APs and DLLs for Windows
<code>\data</code>	: Sample data for APs
<code>\app</code>	: Build environment for APs *
<code>\rtknavi</code>	: RTKNAVI (GUI) *
<code>\strsvr</code>	: STRSVR (GUI) *
<code>\rtkpost</code>	: RTKPOST (GUI) *
<code>\rtkpost_mkl</code>	: RTKPOST_MKL (GUI) *
<code>\rtkplot</code>	: RTKPLOT (GUI) *
<code>\rtkconv</code>	: RTKCONV (GUI) *
<code>\srctblbrows</code>	: NTRIP source table browser (GUI) *
<code>\rtkrcv</code>	: RTKRCV (console) *
<code>\rnx2rtkp</code>	: RNX2RTKP (console) *
<code>\pos2kml</code>	: POS2KML (console) *
<code>\convbin</code>	: CONVBIN (console) *
<code>\str2str</code>	: STR2STR (console) *
<code>\appcmn</code>	: Common routines for GUI APs *
<code>\icon</code>	: Icon data for GUI APs *
<code>\mkl</code>	: Intel MKL libraries for Borland environment *
<code>\test</code>	: Test program and data *
<code>\util</code>	: Utilities *
<code>\doc</code>	: Document files

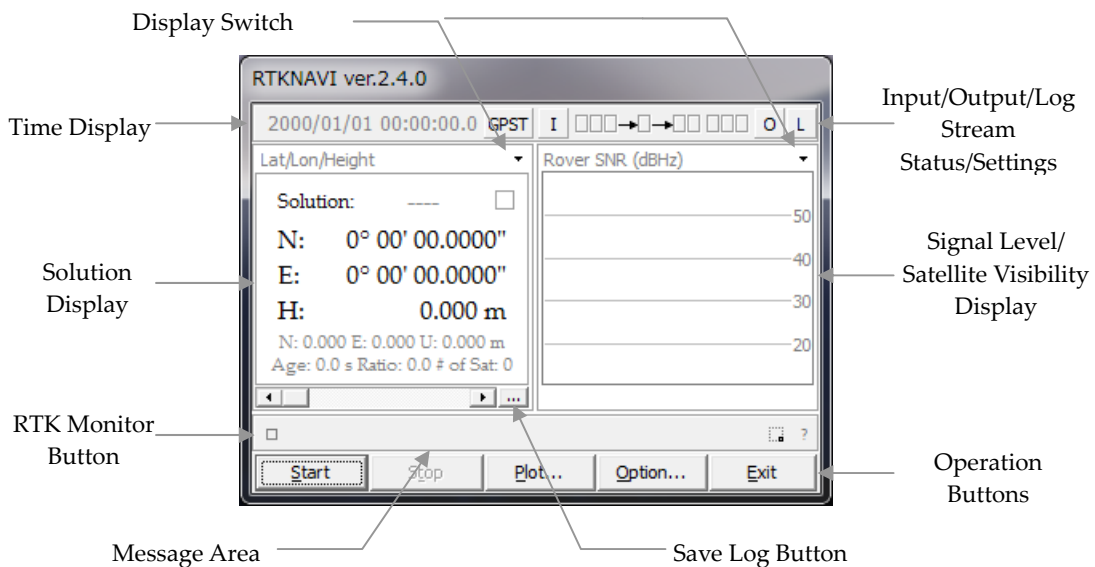
\* Not included in the package `rtklib_<ver>_bin.zip`

- (2) Create the shot-cuts of the GUI AP executable binaries in `rtklib_<ver>\bin`. To execute console APs, add `<install dir>\rtklib_<ver>\bin` to the command path.

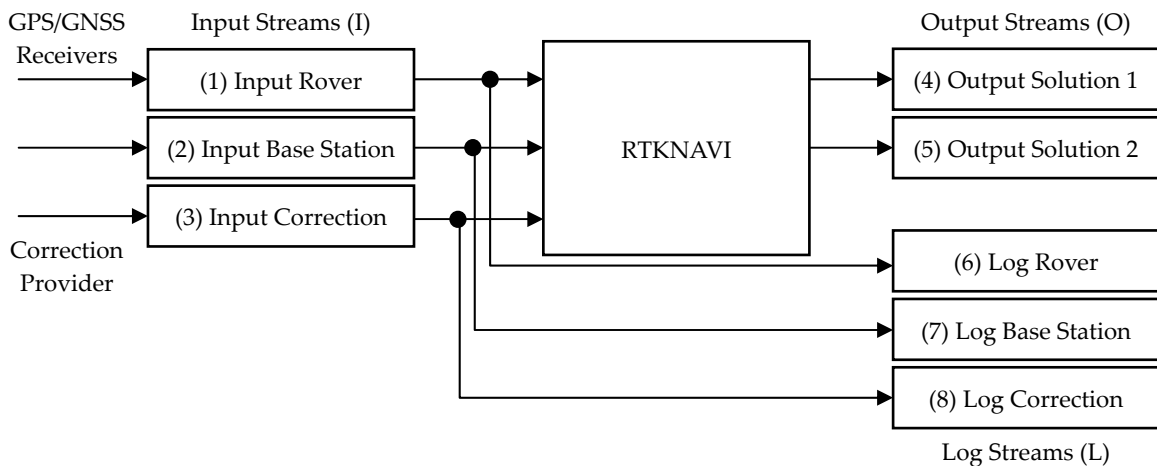
### 3.2 Real-time Positioning with RTKNAVI

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 positioning is enabled with OTF (On-the-fly) integer ambiguity resolution.

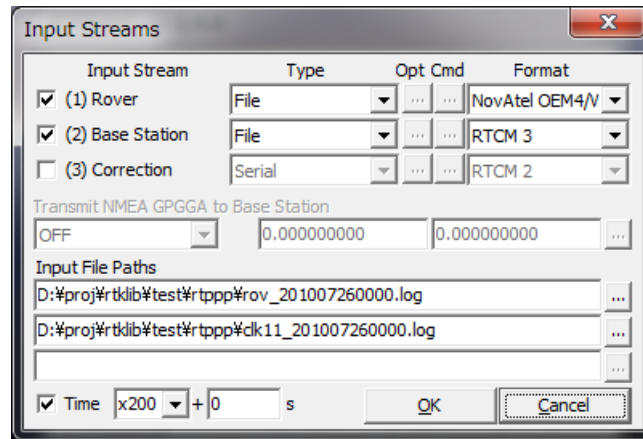
- (1) Execute the binary AP file `rtklib_<ver>\bin\rtknavi.exe`. (Double-click the icon or full in the path in the command console) You can see the main window of RTKNAVI.



- (2) The following figure shows the data flow of RTKNAVI. You have to set Input Streams, Output Streams (optional) and Log Streams (optional) for real-time positioning.



- (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 "Input Streams" dialog.



- (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 stream for Base-station and Correction are not required. The stream types can be selected from the following options.

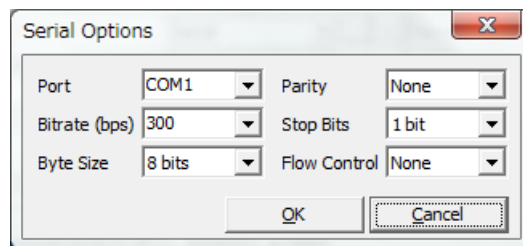
- (a) Serial : Input data from a serial port (RS232C or USB)
- (b) TCP Client : Connect to a TCP server and input data via the TCP connection
- (c) TCP Server : Accept a TCP client connection and input data via the TCP connection
- (d) NTRIP Client : Connect to a NTRIP server (caster) and input data 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 Release Notes for the detailed supported messages.

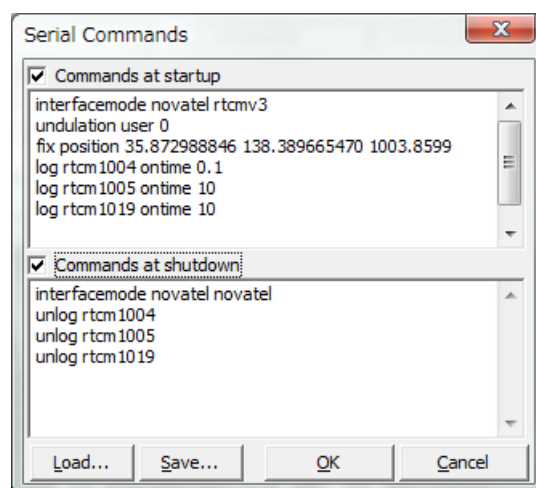
- (a) RTCM2 : RTCM 2.3
- (b) RTCM3 : RTCM 3.1
- (c) NovAtel OEM4/V : NovAtel OEM4/V, OEMStar binary format
- (d) NovAtel OEM3 : NovAtel OEM3 (Millennium) binary format
- (e) u-blox : u-blox LEA-4T/5T binary format

- (f) Superstar II : NovAtel Superstar II binary format
- (g) Hemisphere : Hemisphere Crescent/Eclipse binary format
- (h) SkyTraq : SkyTraq S1315F binary format
- (i) 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 "Serial Options" dialog.



(6) In case of selecting Serial as the stream type, you can configure the startup and shutdown commands to be sent to the GPS/GNSS receiver. Push [...] button under "Cmd" label. Fill in the commands in the text fields in "Serial 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 [Load...] button or save the commands to a command file with [Save...] button. The command file is just a text file including startup commands and shutdown commands separated by a line starting with "!". The sample command files for some typical receivers are found in the directory `rtklib_<ver>/data`.



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



!UBX ... : u-blox LEA-4T/5T command

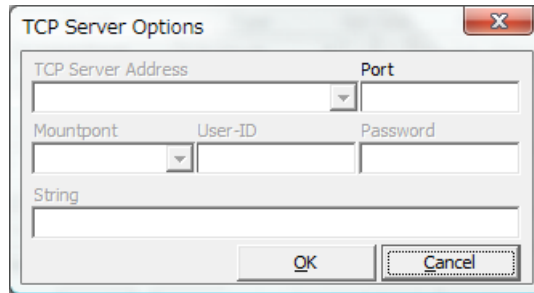
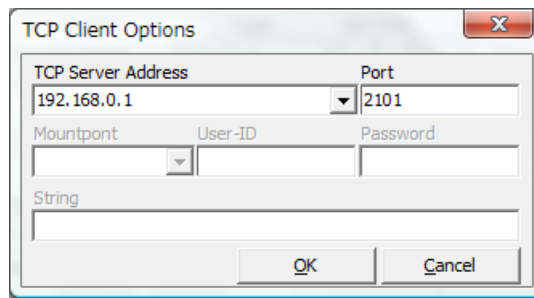
!UBX CFG-PRT portid res0 res1 mode baudrate inmask outmask flags  
 !UBX CFG-USB vendid prodid res1 res2 power flags vstr pstr serino  
 !UBX CFG-MSG msgid rate0 rate1 rate2 rate3  
 !UBX CFG-NMEA filter version numsv flags  
 !UBX CFG-RATE meas nav time  
 !UBX CFG-CFG clear\_mask save\_mask load\_mask  
 !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

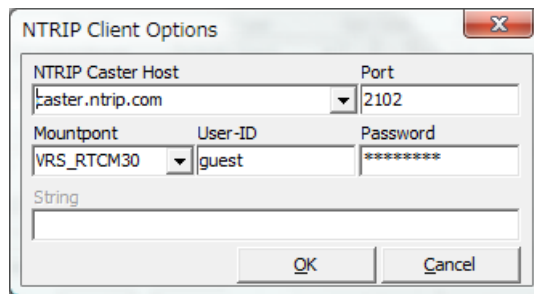
!STQ RESTART [arg...] system restart  
 !STQ CFG-SERI [arg...] configure serial port property  
 !STQ CFG-FMT [arg...] configure output message format  
 !STQ CFG-RATE [arg...] configure binary measurement output rates

!WAIT time : wait for time (ms)

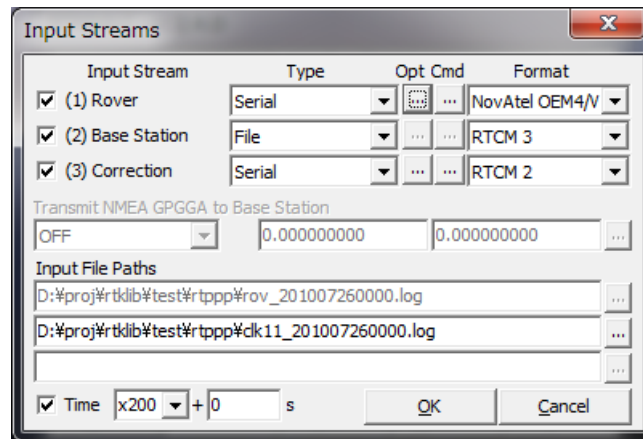
- (8) If you select TCP Client or TCP Server as the stream type, you can set the options of TCP server address (for TCP Client only) and port number with "TCP Client Options" or "TCP Server Options" dialog. If you select "TCP Server", multiple TCP client connections are allowed.



- (9) If you select NTRIP Client as the stream type, you can set the NTRIP client options of NTRIP caster host address, port number, mount-point of NTRIP caster, user-ID and password with "NTRIP Client Options" dialog.



- (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.)

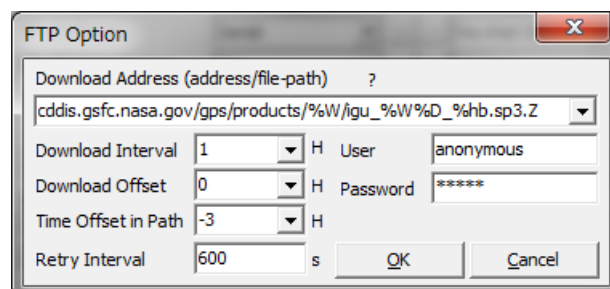


(11) In case of using NRTK (network RTK) service, which requires NMEA GPGGA messages to select reference station or to get virtual reference station position, select the message content with the pull down menu at Transmit NMEA GPGGA to Base/NRTK. If you select Latitude/Longitude to send fixed position, fill in the latitude and longitude for NMEA GPGGA message in degree (minus means south or west).


(12) For correction, you can select FTP or HTTP for the stream type. After pushing Opt button, you have to configure FTP or HTTP options with FTP Option or HTTP Option dialog. At first fill in the server address and the file path in Download Address field as the format <server address>/<file path>. Usually you might include day or time keywords in <file path>. For example, if downloading IGS Ultra-rapid ephemeris from NASA GSFC CDDIS data server, you can input the download address:

**cddis.gsfc.nasa.gov/gps/products/%W/igu\_%W%D\_%hb.sp3.Z**

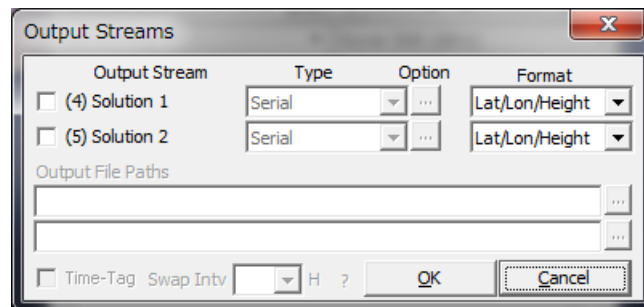
In this case, the keywords %W, %D and %hb are replaced by GPS week number, day of week and 6 hour of the day according to the download time in GPS Time, respectively. For other keywords which can be used in the file path, push  button and see the dialog. You can also set Download Interval, Download Offset (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 the anonymous FTP server.




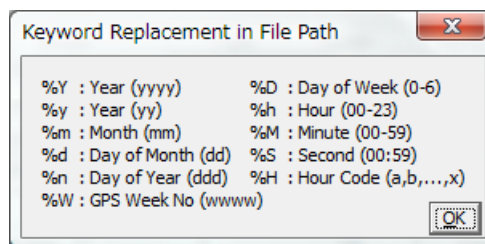
To use download files, you also have to set the file format in Input Dialog. Current version only supports SP3 precise ephemeris for this purpose. Downloaded files are saved in a local directory. The local directory path can be set with Options dialog - Files - FTP/HTTP Local Directory.

- (13) To output of the positioning solution of RTKNAVI, set the output streams. To set the output streams, push the button  upper right in the main window. You can see "Output Streams" dialog. Check and set the stream type of solution in the dialog. You can configure two independent output streams. 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 in output messages can be configured by the positioning options.

- (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) NMEA-183 : NMEA GPRMC, GPGGA, GPGSV

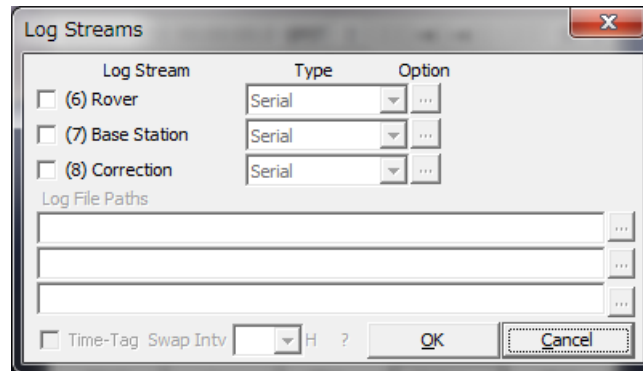


- (14) If you select "File" as the output stream type, you can include some keywords in the file path to be replaced by date or time. Push  button to show the keyword replacement in file paths. If you set Swap Intv option, the output file is swapped periodically in the specified cycle. To use the file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

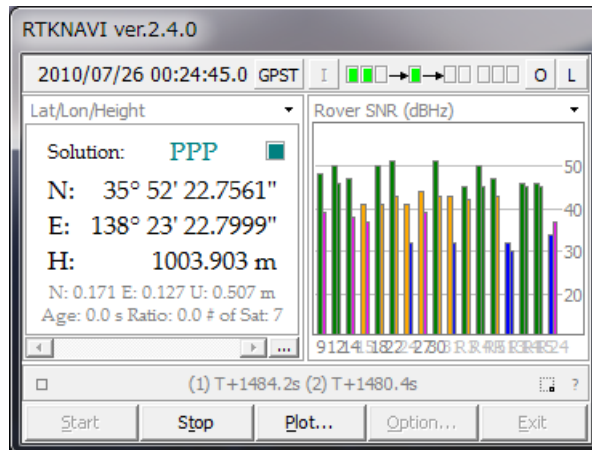


- (15) To output an input stream as a path-through log, set the log streams. To configure the log streams,

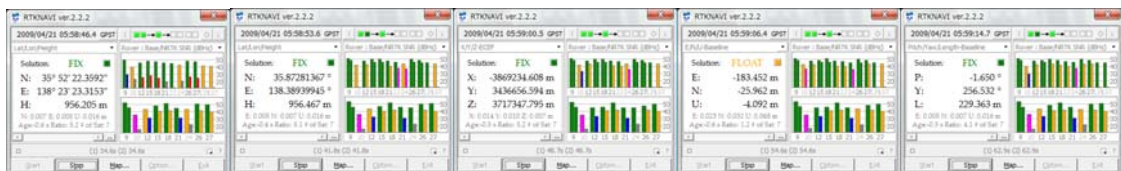
push the button **L** upper right in the main window. You can see "Log Streams" dialog. The settings are similar to the output streams. If you want to replay the log file as an input stream later, you have to check "Time-Tag" option and output the time tag file simultaneously. The output path of the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



- (16) To configure the positioning options, push **Options...** button and set the options in "Options" dialog. For details of the positioning options, refer 3.5 Configure Positioning Options for RTKNAVI and RTKPOST.
- (17) 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.



(18) 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  $\nabla$  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.

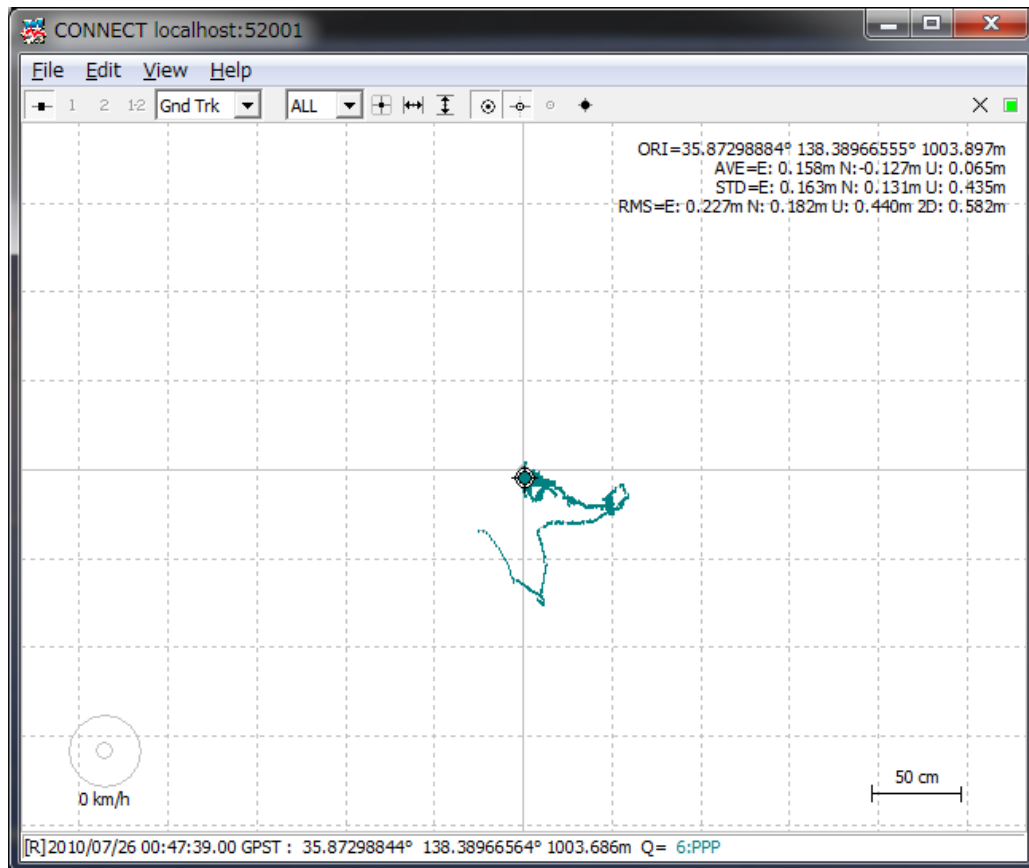


(19) To switch the format in the time display, push **GPST** button upper center in the main window. You can switch the time system to GPST, UTC, JST and GPST (GPS week/TOW), alternatively.

(20) In the status display right in the main window, observation SNR status or visible satellites in skyplot are shown. By pushing  $\nabla$  button upper right corner of the status display, you can switch the contents to Rover : Base/NRTK SNR, Rover SNR, Rover Skyplot, Base/NRTK Skyplot and Baseline plot, alternatively.




- (21) By pushing  button, you can execute RTKPLOT to plot the current position of the rover receiver as the graph. For the usage of RTKPLOT, please refer 3.7 View and Plot Solutions and Observation Data with RTKPLOT.



- (22) 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 "Options" dialog.
- (23) By pushing the  button lower left in the main window, you can see "RTK Monitor" window. With the window, you can monitor 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 close the window, push  button.

Parameter	Value
RTKLIB Version	2.2.0
RTK Server Thread	400
RTK Server State	Run
Processing Cycle (ms)	10
Input Buffer Size (bytes)	32768
Positioning Mode	Single
Frequencies	L1
Elevation Mask (deg)	15
SNR Mask (dbHz)	35
SBAS Corrections	On,On,On,Off
Data in Input Buffer (bytes)	0,0
Accumulated Time to run	00:03:11.1
# of Input Data Rover	190,8,0,382,0,0
# of Input Data Base Station	0,0,0,0,0,0
Solution Status	SBAS
Time of Receiver Clock Rover	2009/01/29 02:00:30.000000081

- (a) RTK : General status of the internal positioning process
- (b) Satellites : Status of each satellites
- (c) States : State vector values of the estimation filter
- (d) Covariance : Covariance matrix of the estimation filter
- (e) Obs Data : Input observation data. RCV=1 means rover and 2 means base-station
- (f) Nav GPS/GAL/QZS : GPS, Galileo or QZSS navigation messages
- (g) Nav GLONASS : GLONASS navigation messages
- (h) Nav GEO : SBAS satellites navigation messages
- (i) Streams : Status of input, output and log streams
- (j) SBAS Msgs : HEX dump of input SBAS messages
- (k) SBAS Long : SBAS long term satellite corrections
- (l) SBAS Iono : SBAS ionospheric delay corrections
- (m) SBAS Fast : SBAS fast corrections
- (n) RTCM Msgs : Status of RTCM 2 or 3 messages
- (o) RTCM DGPS : RTCM DGPS corrections
- (p) RTCM SSR : RTCM SSR corrections
- (q) (1) Rover : Dump of Input Rover stream
- (r) (2) Base/NRTK : Dump of Input Base Station stream
- (s) (3) Correction : Dump of Input Correction stream
- (t) (4)(5)Solution 1/2 : Dump of Output Solution 1/2 stream
- (u) Error/Warning : Error or warning messages

(24) By pushing  button lower right in the main window, you can minimize the main window as an icon



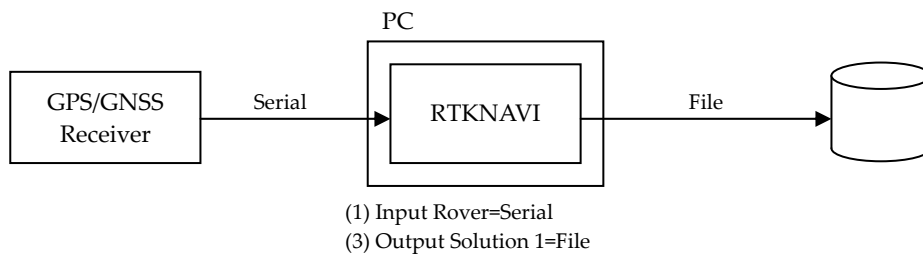
in the task-tray of Windows desktop. To restore the main window, double-click the task-tray-icon or click right-button on the task-tray-icon and select menu "Main Window...".

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

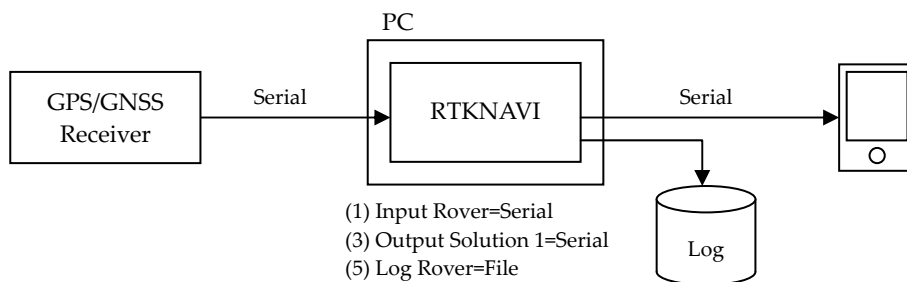
For relative positioning like RTK-GPS/GNSS, the rover receiver and the base station receiver are usually placed on the separated sites. In other cases, user might use the positioning result at the remote site far from these receivers. To interconnect these sites, user has to establish data communication links. To facilitate easy setup of these communication links, RTKLIB provides the communication utility 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.

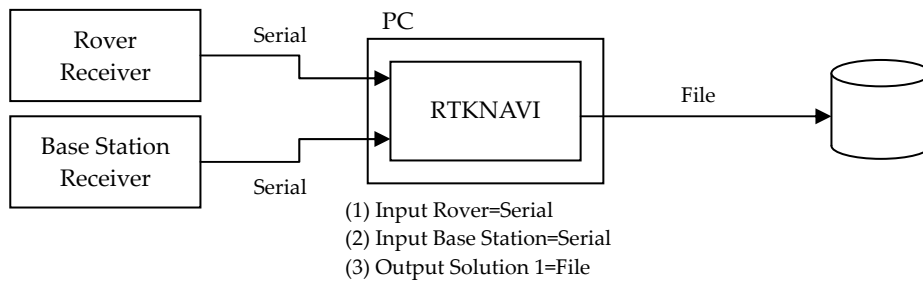
- (1) Single-point positioning and output solutions to a file



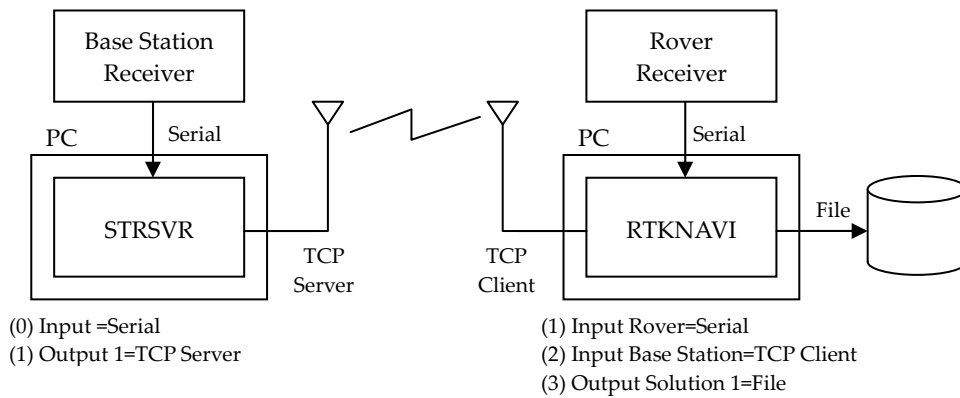
- (2) Single point positioning, output solutions to a serial device, log data to a file



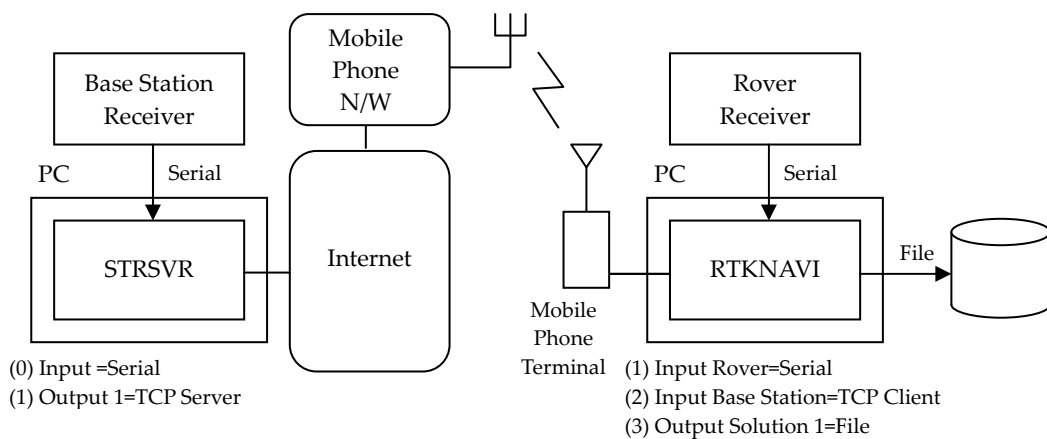
- (3) RTK-GPS/GNSS, input the rover and base-station data from two serial devices



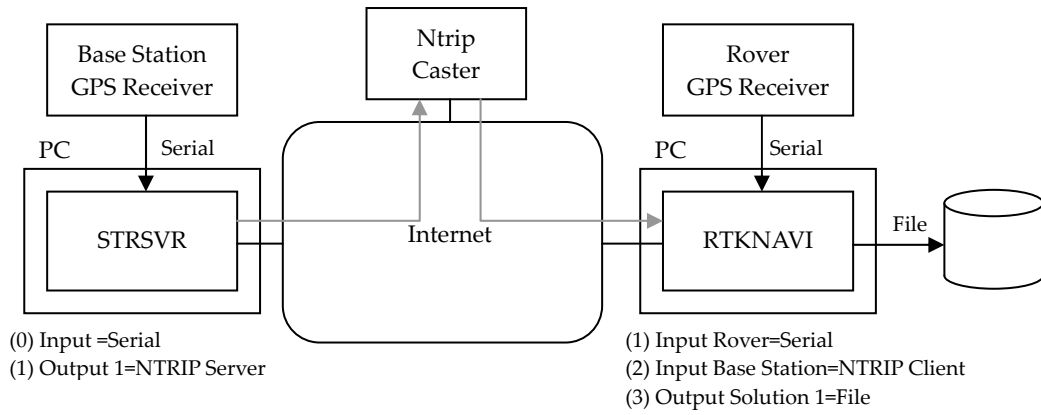
- (4) RTK-GPS/GNSS, input rover data from a serial port and input base-station data from a remote receiver via WiFi network.



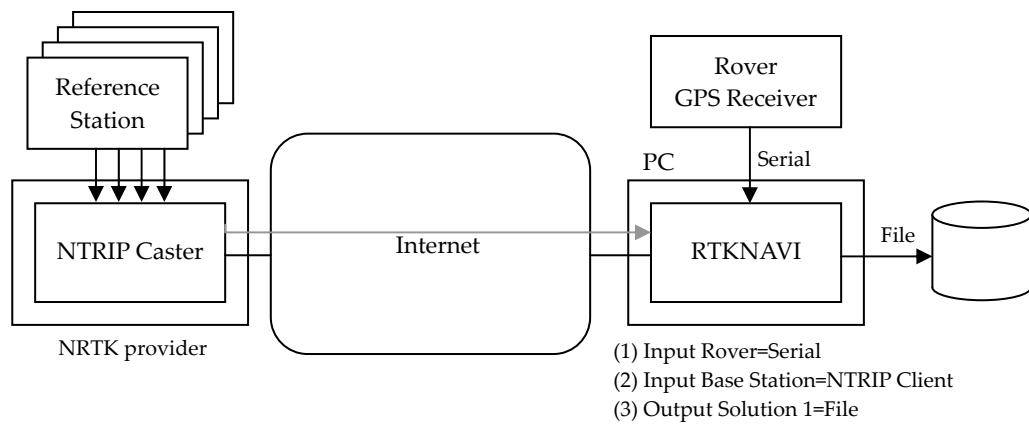
- (5) RTK-GPS/GNSS, input rover data from a serial port and input base station data via mobile phone Internet connection



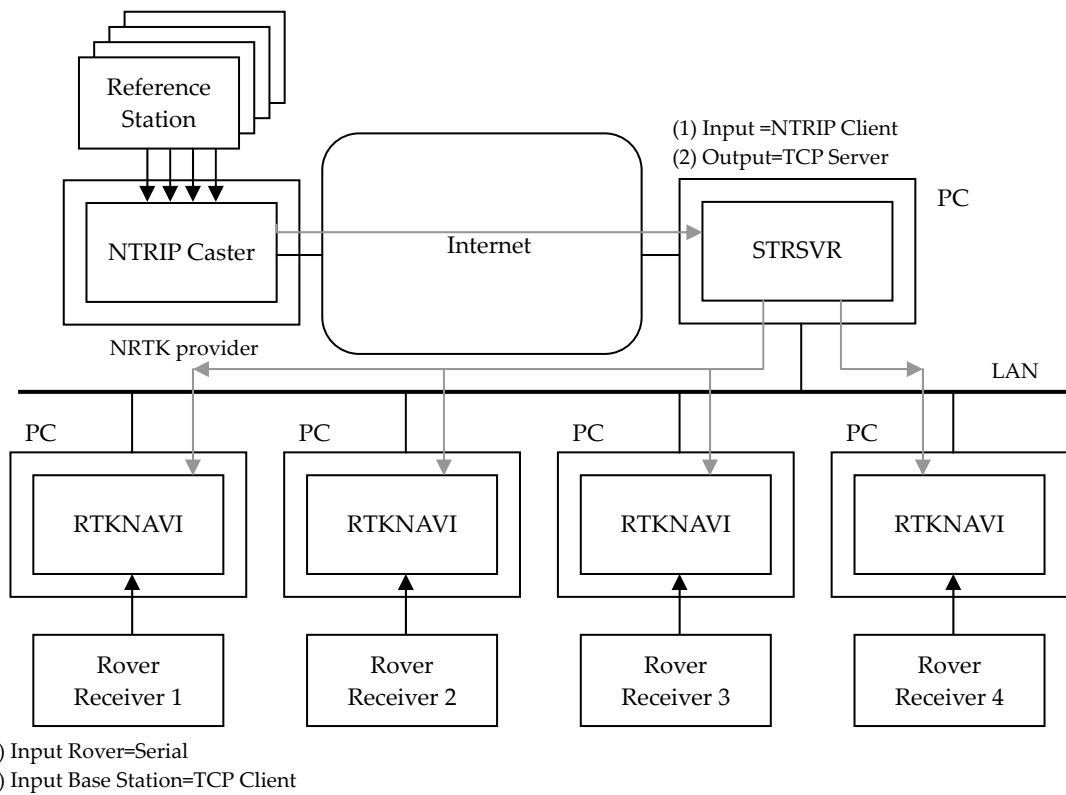
(6) RTK-GPS/GNSS, input data from a serial port and input base station data via NTRIP caster on Internet



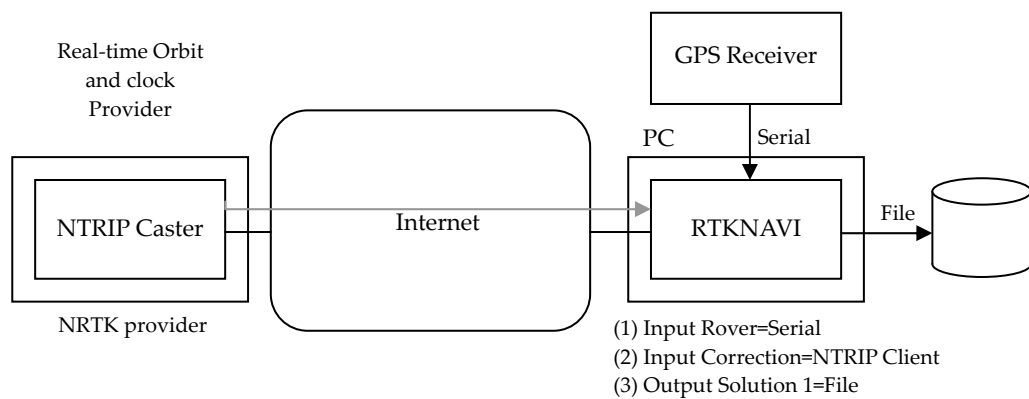
(7) RTK-GPS/GNSS with NRTK (Network RTK) service via Internet



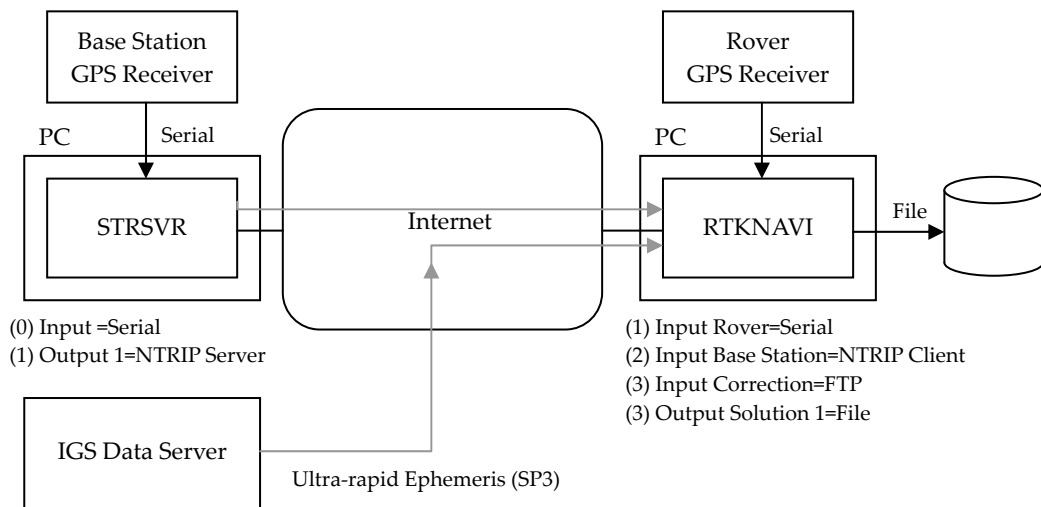
(8) Multiple RTK-GPS/GNSS with single NRTK service



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

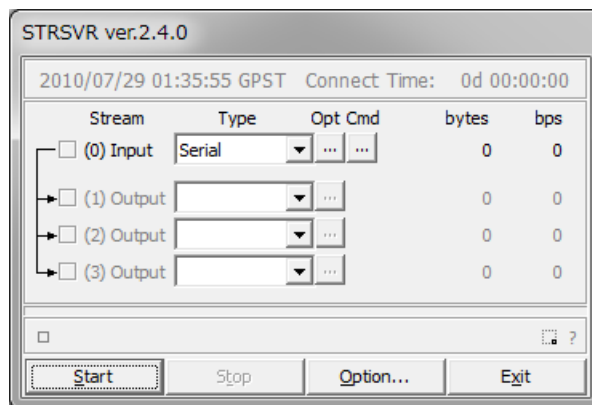


(10) Long-baseline RTK with FTP download of precise ephemeris



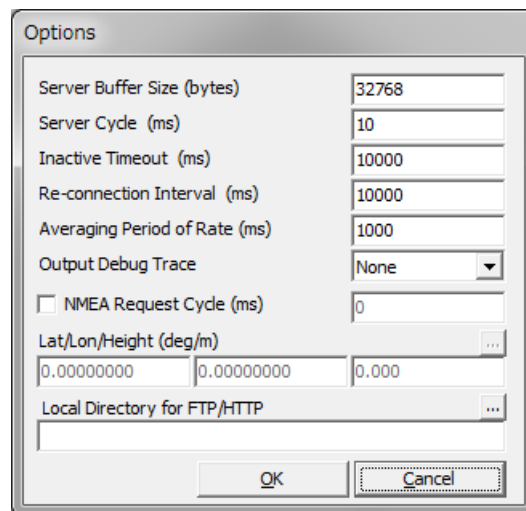
The following instructions are for the operation of STRSVR.

- (1) Execute the binary AP file `rtklib_<ver>\bin\strsvr.exe`. You can see the 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, File, FTP or HTTP. 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 same as Output Streams or Log Streams for RTKNAVI.

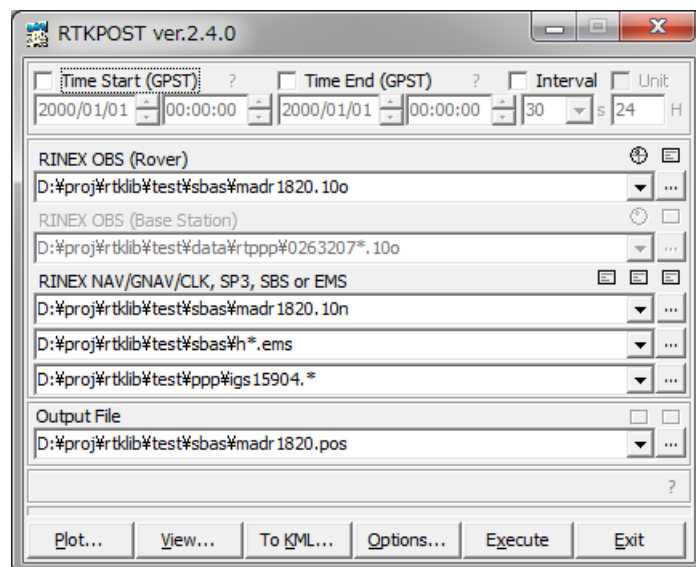
- (4) Push **Start** button in the main window. The communication status is shown in the message area lower center of the main window. Status indicators left side of the main window also shows the communication status. The indicator colors means: **Orange**: waiting connection, **Dark-Green**: connected, **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.
- (5) By pushing **Options...** button, you can set the communication options with "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.



### 3.4 Post-Processing Analysis with RTKPOST

RTKLIB contains a post processing analysis AP RTKPOST. RTKPOST inputs the standard RINEX 2.10 or 2.11 observation data and navigation message files (GPS and GLONASS) and can compute the positioning solutions by carrier-based relative positioning.

- (1) Execute the binary AP file `rtklib_<ver>\bin\rtkpost.exe`. You can see the main window of RTKPOST. You can execute the binary AP file `rtklib_<ver>\bin\rtkpost_mkl.exe` instead. RTKPOST\_MKL is a faster version of RTKPOST, which link Intel MKL library for fast matrix computation.

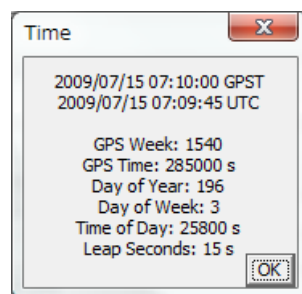


- (2) Input the RINEX observation data file path of the rover receiver in the text field RINEX OBS (Rover). Fill in the file path or select a file using the file selection dialog shown by pushing  button. You can use the compressed file by GZIP (`.gz`), COMPRESS (`.z`) or Hatanaka-Compression (`.yyd`) for the RINEX observation data. If a wild-card (`*`) is included in the file path, the wild-card expanded multiple files are used.
- (3) If you process RINEX data in the positioning mode of DGPS/DGNSS, Kinematic, Static, Moving-Base or Fixed, input the file path of the base-station receiver in RINEX OBS (Base Station) field as well as rover observation data.
- (4) You also have to input the path of RINEX navigation message files of GPS and GLONASS in RINEX



NAV/GNAV/HNAV/CLK, SP3, SBS or EMS field. If you leave first and second field blank, the observation data file path with the extension replaced by **.nav and .gnav (.obs)** or **.yyg and .yyg** (**.yyo**) is used for the navigation message files of GPS and GLONASS. If a wild-card (\*) is included in the file path, the wild-card expanded multiple files are used. To use precise ephemeris and clock for PPP-kinematic or PPP-static mode, you can input SP3-c (for precise satellite ephemeris and clock) or RINEX CLK (for precise satellite clock) file path in the field. For SBAS correction, you can input SBAS message log 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.

- (5) Input the output file path in the text field Output File.
- (6) Push  button to set the processing options. For detailed options, 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 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.



- (7) If you check both of Time Start and Time End, you can check Unit for multiple session analysis. If Unit checked and set the Unit, the analysis session is separated to multiple sessions for the Unit (hr). To avoid overwriting the output to the same file, the output file path has to contain the keyword replaced according to the session time. For details of the keyword replacement in the input or output file paths, refer 3.5 Configure Positioning Options for RTKNAVI and RTKPOST.
- (8) Push  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  button.
- (9) After completing the analysis, by pushing  button, you can display the content of the output file by Text Viewer. You can reload the output file by pushing  button in Text Viewer window. To

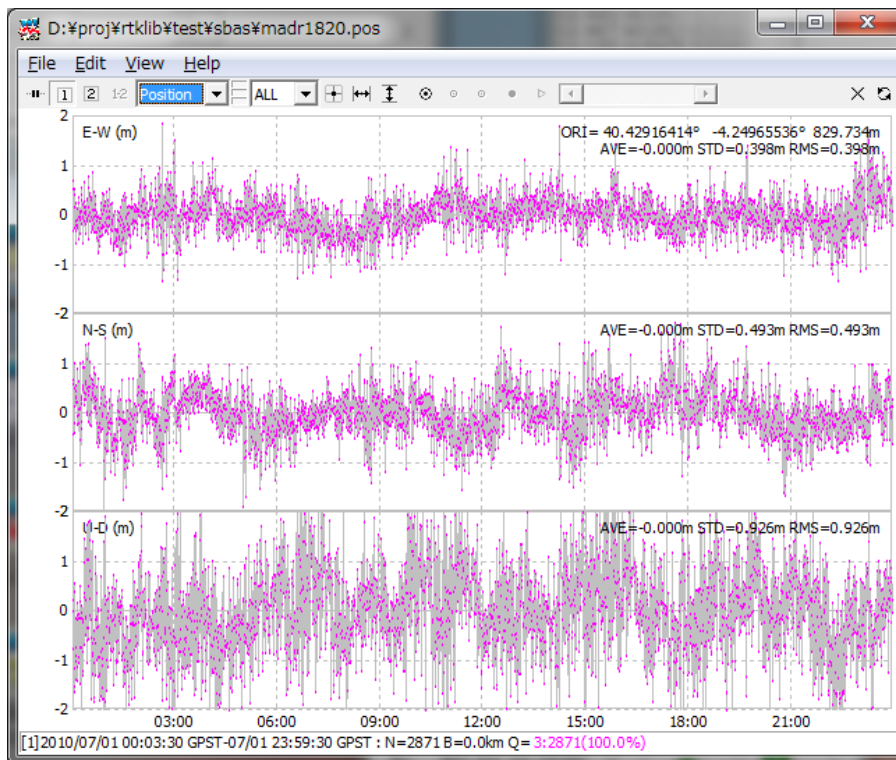
close the window, push **Close** button.

```

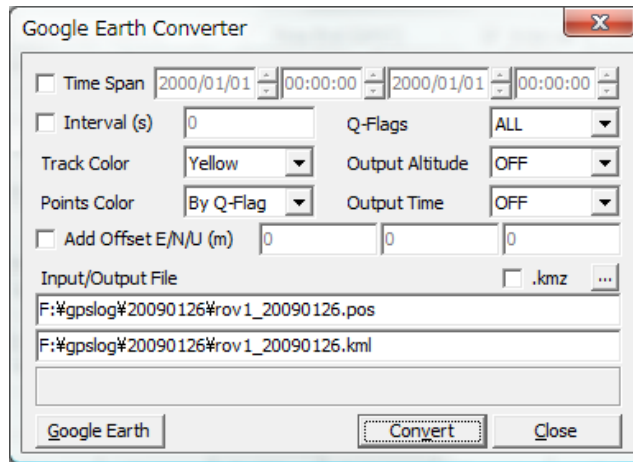
D:\proj\rtklib\test\sbas\madr1820.pos
Read... Option... Close
# program : RTKPOST ver.2.4.0
# inp file : D:\proj\rtklib\test\sbas\madr1820.10o
# inp file : D:\proj\rtklib\test\sbas\madr1820.10n
# inp file : D:\proj\rtklib\test\sbas\h*.ems
# inp file : D:\proj\rtklib\test\ppp\igs15904.*
# obs start : 2010/07/01 00:00:00.0 GPST (week1590 345600.0s)
# obs end : 2010/07/01 23:59:30.0 GPST (week1590 431970.0s)
# pos mode : single
# elev mask : 10.0 deg
# snr mask : 0.0 dBHz
# ionos opt : sbas
# tropo opt : sbas
# ephemeris : broadcast+sbas
# navi sys : gps sbas
#
# (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) sdr
2010/07/01 00:03:30.000 40.429170249 -4.249659068 828.9639 3 8 1.2508 0.0
2010/07/01 00:04:00.000 40.429160404 -4.249649049 829.7288 3 8 1.2462 0.0
2010/07/01 00:04:30.000 40.429169389 -4.249646984 829.9714 3 9 1.1564 0.0
2010/07/01 00:05:00.000 40.429160664 -4.249650521 830.2922 3 9 1.1411 0.0
2010/07/01 00:05:30.000 40.429165454 -4.249658182 828.1121 3 9 1.1494 0.0
2010/07/01 00:06:00.000 40.429173361 -4.249658278 828.2099 3 9 1.1440 0.0
2010/07/01 00:06:30.000 40.429168610 -4.249653364 830.1209 3 9 1.1536 0.0
2010/07/01 00:07:00.000 40.429162484 -4.249648980 828.5924 3 9 1.1390 0.0



```

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



- (11) By pushing **To KML...** button, the output file can be converted to Google Earth KML file with "Google Earth Converter" dialog. Set or select the options and push **Convert** button in the dialog.

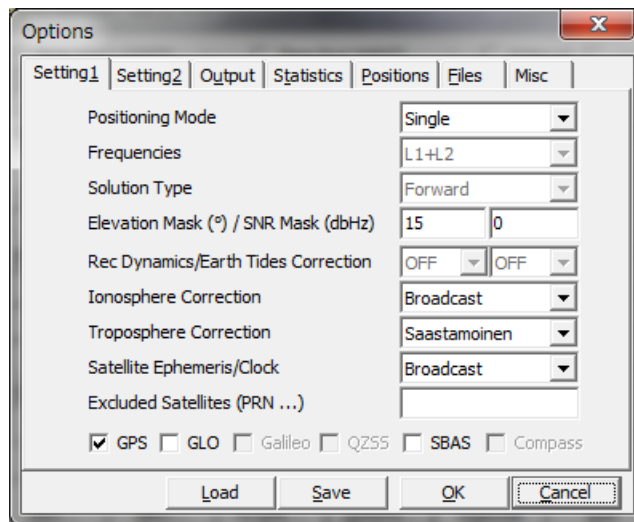


- (12) With  button in the main window, you can view and plot the input observation data RTKPLOTT. You can also display the contents of the input files with Text Viewer by pushing  button.

### 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.

(1) Setting 1

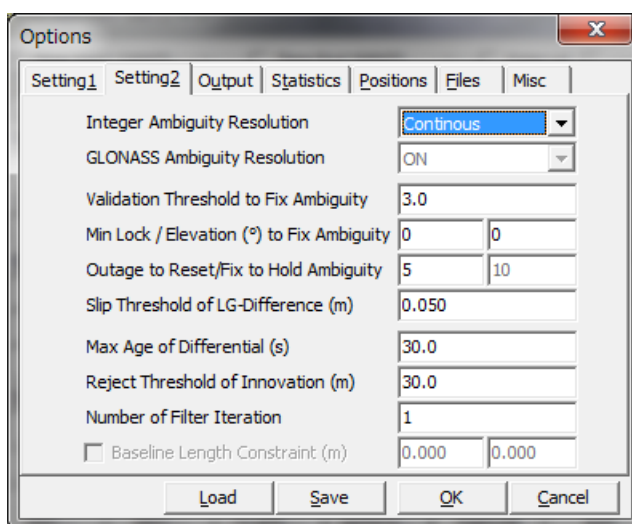


Item	Descriptions	Notes
Positioning Mode	Set positioning mode - Single : Single point positioning or SBAS DGPS - DGPS/DGNSS : Code-based differential GPS - Static : Carrier-based Static positioning - Kinematic: Carrier-based Kinematic positioning - Moving-Base: Moving baseline - Fixed: Rover receiver position is fixed - PPP Kinematic: Precise Point Positioning with kinematic mode - PPP Static: Precise Point Positioning with static mode	
Frequencies	Set used carrier frequencies - L1 : L1 only - L1+L2 : L1 and L2 - L1+L2+L5: L1, L2 and L5	

Item	Descriptions	Notes
Solution Type	Set solution type - Forward: Forward filter solution - Backward: Backward filter solution * - Combined: Smoother combined solution with forward and backward filter solutions *	* RTKPOST only
Elevation Mask	Set elevation mask angle (deg)	
SNR Mask	Set SNR mask (dBHz)	
Rec Dynamics	Set the dynamics model of the rover receiver. - OFF: Dynamics is not used - ON: Receiver velocity and acceleration are estimated. The receiver position is predicted with the estimated velocity and acceleration.	Only effective in DGPS/DGNSS or Kinematic mode
Earth Tides Correction	Set whether earth tides correction is applied or not - OFF: Not apply earth tides correction - ON: Apply earth tides correction	Only available in DGPS/DGNSS, Kinematic or PPP Mode
Ionosphere Correction	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. - OFF : Not apply ionospheric correction - Broadcast: Apply broadcast ionospheric model - SBAS: Apply SBAS ionospheric model - Dual-Frequency: Ionosphere-free linear combination with dual frequency measurements is used for ionospheric correction - Estimate STEC : Estimate ionospheric parameter STEC (slant total electron content)	Estimated is effective for only DGPS/DGNSS or Kinematic, Static mode.
Troposphere Correction	Set whether tropospheric parameters (zenith total delay at rover and base-station positions) are estimated or not. - OFF : Not apply troposphere correction - Saastamoinen: Saastamoinen model - SBAS: SBAS tropospheric model - Estimate ZTD: Estimate ZTD (zenith total delay) parameters - Estimate ZTD+Grad: Estimate ZTD and horizontal gradient parameters	
Satellite Ephemeris	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 SSR correction (antenna phase center value) - Broadcast+SSR CoM: Broadcast ephemeris with SSR correction (satellite center of mass value)	SSR corrections can be applied for RTKNAVI
Excluded Satellites (PRN)	Set the excluded satellites for positioning. Fill in the PRN numbers of the satellites separated by spaces.	

Item	Descriptions	Notes
Navi System	<p>Check used navigation satellite systems. If not checked, satellites of the system are not used for positioning.</p> <ul style="list-style-type: none"> <li>- GPS</li> <li>- GLONASS</li> <li>- Galileo *</li> <li>- QZSS *</li> <li>- SBAS</li> <li>- Compass *</li> </ul>	* Not supported yet

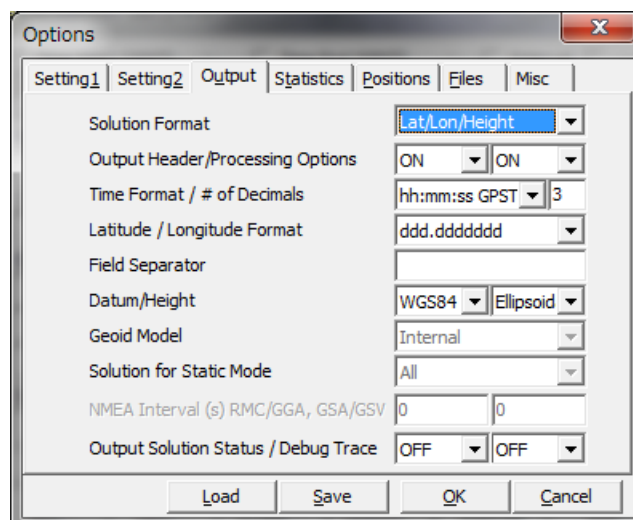
(2) Setting 2



Item	Descriptions	Notes
Integer Ambiguity Resolution	<p>Set the strategy of integer ambiguity resolution</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 held as the resolved values.</li> </ul>	
GLONASS Ambiguity Resolution	<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. The different receiver types have inter-channel bias which can not be canceled by DD.</li> <li>- Auto calibration: Receiver inter-channel bias terms are estimated as a linear equation by the frequencies.</li> </ul>	Default value: 3.0
Validation Threshold of AR	<p>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.</p>	Default value: 3.0

Item	Descriptions	Notes
Min Lock to Fix Ambiguity	Set the lock count to fix integer ambiguity. If the lock count is less than the value, the ambiguity is excluded from the fixed integer vector.	Default value: 0
Min Elevation to Fix Ambiguity	Set the minimum elevation angle (deg) to fix ambiguity. The ambiguities of the satellites under the setting are kept as float values.	Default value: 0
Outage to Reset Ambiguity	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.	Default value: 5
Fix to Hold Ambiguity	If the integer ambiguity resolution strategy is set to "Fix and Hold", set the number of continuous fixes to hold the ambiguity.	Default value: 10
Slip Threshold of LG-Difference	Set the cycle-slip threshold (m) of geometry-free LC carrier-phase difference between epochs.	Default value: 0.05
Max Age of Differential	Set the maximum value of age of differential (s) between the rover and the base station.	Default value: 30
Reject Threshold of Innovation	Set the reject threshold of innovation (pre-fit residual) (m). If the innovation is over the value, the observable is excluded for estimation as an outlier.	Default value: 30
Number of Iteration	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 is effective to handle the nonlinearity of measurement equation.	Default value: 1
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.	

(3) Output



Item	Descriptions	Notes
------	--------------	-------

Item	Descriptions	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, GPGSV	For RTKNAVI, specify options as Output Streams setting.
Output Header	Set whether the header is output or not.	Not applicable to NMEA
Output Processing Options	Set whether the processing options are output or not.	RTKPOST only Not applicable to NMEA
Time Format	Set the format of time - ssssssss.sss GPST : GPS week and TOW (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	Not applicable to NMEA
# of Decimals	Set number of decimals in the time format	Not applicable to NMEA
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	Not applicable to NMEA
Field Separator	Set the separator for fields.	Not applicable 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	
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.	
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 outputted. The time of solution is first epoch in the processing period.	RTKPOST only
NMEA Interval (s) RMC/GGA	Set the output interval of NMEA GPRMC, GPGGA messages	RTKNAVI only
NMEA Interval (s) GSA/GSV	Set the output interval of NMEA GPGSA, GPGSV messages	RTKNAVI only



Item	Descriptions	Notes
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).	
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).	

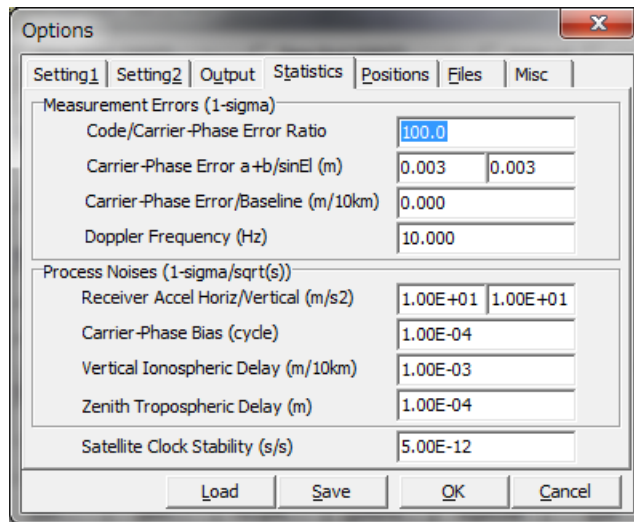
\*1 WW15MGH.DAC (<http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm96/binary/binarygeoid.html> )

\*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> )

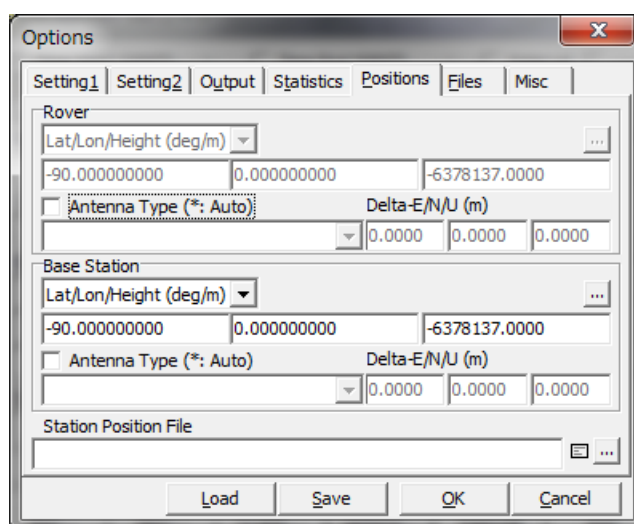
(4) Statistics



Item	Descriptions	Notes
Measurement Errors (1-sigma)		
Code/Carrier-Phase Error Rate	Set the ratio of standard deviations of pseudorange errors to carrier-phase errors.	Default value: 100
Carrier-Phase Error	Set the base term of carrier-phase error standard deviation (m).	Default value: 0.003
Carrier-Phase Error/sinEl	Set the elevation dependent term of carrier-phase error standard deviation (m/sin(el)).	Default value: 0.003
Carrier-Phase Error/Baseline	Set the baseline-length dependent term of carrier-phase error standard deviation (m/10km).	Default value: 0
Doppler Frequency	Set the standard deviation of Doppler errors (Hz) (Current version does not use the value)	Default value: 1

Item	Descriptions	Notes
Process Noises (1-sigma/sqrt(s))		
Receiver Accel Horiz/Vertical	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.	Default value: 1 and 0.1
Carrier-Phase Bias	Set the process noise standard deviation of carrier-phase bias (ambiguity) (cycle/sqrt(s)).	Default value: 1E-4
Vertical Ionospheric Delay	Set the process noise standard deviation of vertical ionospheric delay per 10 km baseline (m/sqrt(s)).	Default value: 1E-3
Zenith Tropospheric Delay	Set the process noise standard deviation of zenith tropospheric delay (m/sqrt(s)).	Default value: 1E-4
Satellite Clock Stability	Set the satellite clock stability(s/s). The value is used for interpolation of base-station observables.	Default value: 5E-12

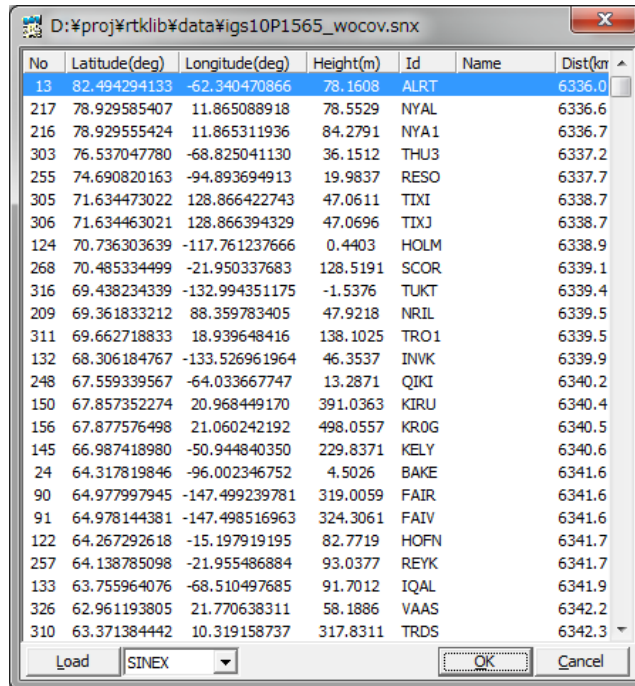
(5) Positions



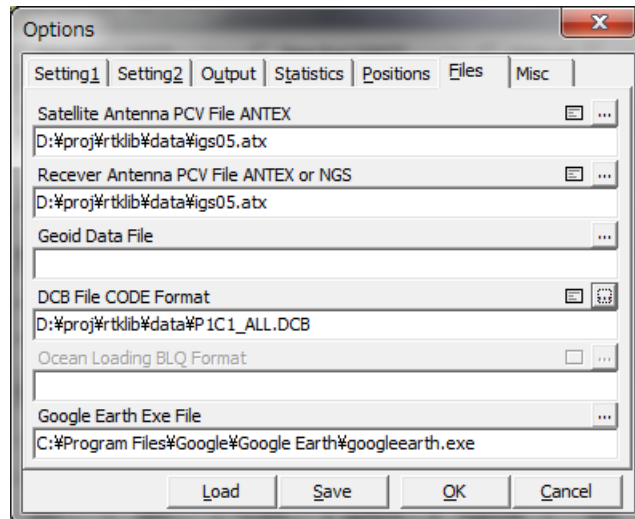
Item	Descriptions	Notes
Rover		
Lat/Lon/Height (deg/m)	Set the position of the rover antenna if the rover antenna is fixed. (current version does not support the option)	
Antenna Type	Select the type of the rover antenna. To select the antenna type, set the Receiver Antenna PCV File path in Files. If "*" is used, antenna type and antenna delta are recognized by the antenna information of RINEX OBS header (RTKPOST) or RTCM antenna information (RTKNAVI).	
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).	

Item	Descriptions	Notes
Base Station		
Lat/Lon/Height (deg/m)	<p>Set the position of the base-station antenna.</p> <ul style="list-style-type: none"> <li>- Lat/Lon/Height (deg/m): Latitude/longitude/height in degree and m</li> <li>- Lat/Lon/Height (dms/m): Latitude/longitude/height in degree/minute/second and m</li> <li>- X/Y/Z-ECEF (m): X/Y/Z components in ECEF frame.</li> <li>- RTCM Station Position: Use the antenna position included in RTCM messages *</li> <li>- Average of Single-Pos: Use the average of single point solutions **</li> <li>- 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. **</li> <li>- RINEX Header Position: Use the approximate position in RINEX OBS header. **</li> </ul>	<p>Height is specified as ellipsoidal height</p> <ul style="list-style-type: none"> <li>* RTKNAVI only</li> <li>** RTKPOST only</li> </ul>
Antenna Type	<p>Select the type of the base-station antenna.</p> <p>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).</p>	
Delta-E/N/U	<p>Set the delta position of the base-station antenna as E/N/U offsets of ARP position with refer to the marker (m).</p>	
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>- Latitude (deg)</li> <li>- Longitude (deg)</li> <li>- Ellipsoidal height (m)</li> <li>- Station ID</li> <li>- Station name</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_wocov.snx</b></p>	

If you set Station Position File in "Files" tab, you can select the portion of the rover or the base-station antenna from the station list in "Stations" dialog by pushing  button. You have to select the format from Lat/Lon/Hgt and SINEX in the dialog.



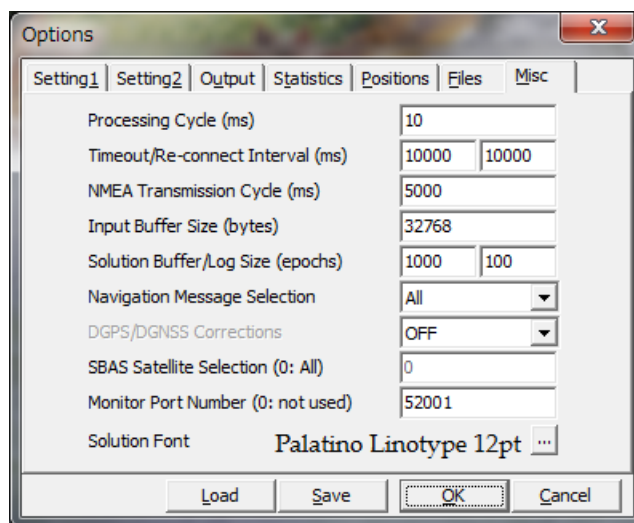
(6) Files



Item	Descriptions	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>igs05.atx</b> file provided by IGS. An example of the ANTEX file is found at <b>rtklib_&lt;ver&gt;\data\igs05.atx</b> .	

Item	Descriptions	Notes
Receiver Antenna PCV File ANTEX or NGS	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\igs05.atx</b> . or <b>rtklib_&lt;ver&gt;\data\ngs_abs.pcv</b> .	
Geoid Data File	Input the file path of the geoid data file if selecting the external model as Geoid Model.	
DCB File CODE Format	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>	
Ocean Loading BLQ Format	(current version does not support ocean loading correction)	
Google Earth Exe File	Input the execution file path of Google Earth.	RTKPOST only
FTP/HTTP Local Directory	Input the local directory for FTP/HTTP download. The downloaded files are save in the directory.	RTKNAVI only

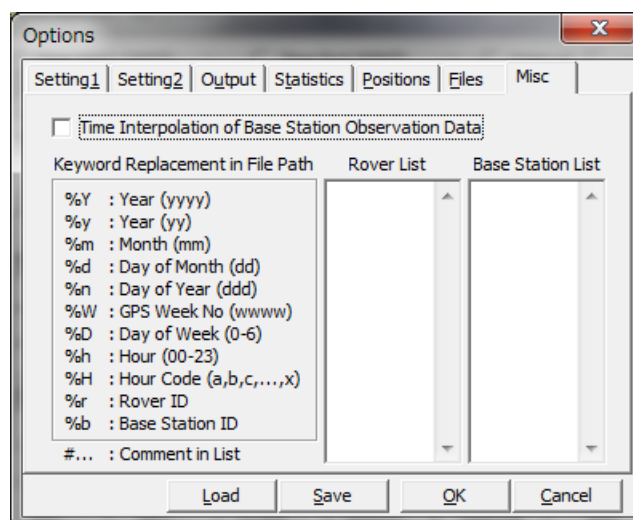
(6) Misc (RTKNAVI)



Item	Descriptions	Notes
------	--------------	-------

Item	Descriptions	Notes
Processing Cycle	Set the processing cycle time of in ms. Usually set 100 ms or less value.	
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.	
NMEA Transmission Cycle	Set the NMEA GPGGA transmission cycle to NRTK server in ms.	
Input Buffer Size	Set the internal input message buffer size in bytes. Usually set it to 32768 or more.	
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.	
Navigation Message Selection	Select navigation message to be used. - Rover: In rover receiver stream - Base Station: In base station receiver stream - Both: In both streams	
DGPS/DGNSS Corrections	(not used in current version)	
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.	
Monitor Port Number	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.	Default: 52001
Solution Font	Select the font of the solution display in the main window.	

## (7) Misc (RTKPOST)



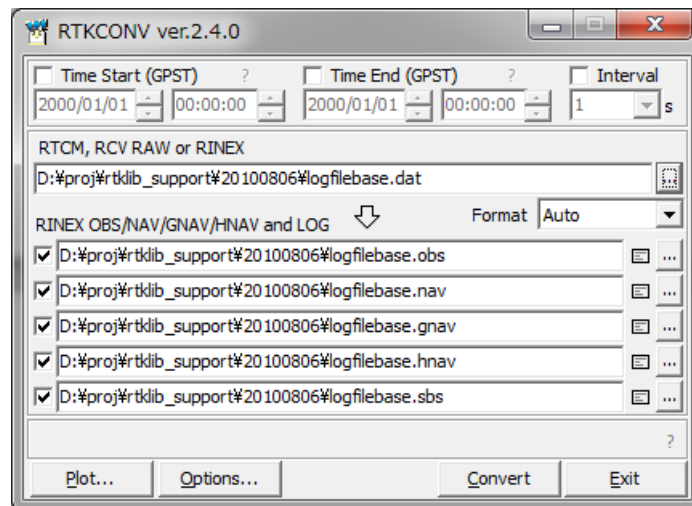
Item	Descriptions	Notes
Time Interpolation of Base Station Observation Data	Check it to enable time interpolation of base station data. If checked, the base station data are linearly interpolated to the rover epoch and DD (double-difference) is made with them. If not checked, nearest epoch of base station data is used for DD.	
Keyword Replacement in File Path	<p>For batch processing with multiple input files or multiple sessions, you can set input file paths or output file path containing the following keywords.</p> <p><b>%Y, %y, %m, %d, %n, %W, %D, %h, %H, %r, %b</b></p> <p>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 <b>%r, %b</b>, input Rover List or Base Station List below.</p>	
Rover List	Input the rover ID list to replace keyword <b>%r</b> in input and output file paths. The line starting with "#" is treated as a comment.	
Base Station List	Input the base station ID list to replace keyword <b>%b</b> 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 data to RINEX OBS (observation data), RINEX NAV (GPS navigation messages) and RINEX GNAV (GLONASS navigation messages). RTKCONV can also extract SBAS messages from the receiver raw data and output the SBAS log file.

The supported RINEX version is 2.10 or 2.11. RINEX 3.0 will be supported by the future version. Refer the Appendix B.2 SBAS Log File for the SBAS log file format.

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



- (2) Input the receiver raw data file path to the text field Log File. 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 Release Notes for detailed supported messages.

- |                    |   |
|--------------------|---|
| (a) RTCM2          | : RTCM 2.3                                |
| (b) RTCM3          | : RTCM 3.0 or 3.1                         |
| (c) NovAtel OEM4/V | : NovAtel OEM4/V binary format            |
| (d) NovAtel OEM3   | : NovAtel OEM3 (Millennium) binary format |
| (e) u-blox         | : u-blox LEA-4T/5T binary format          |



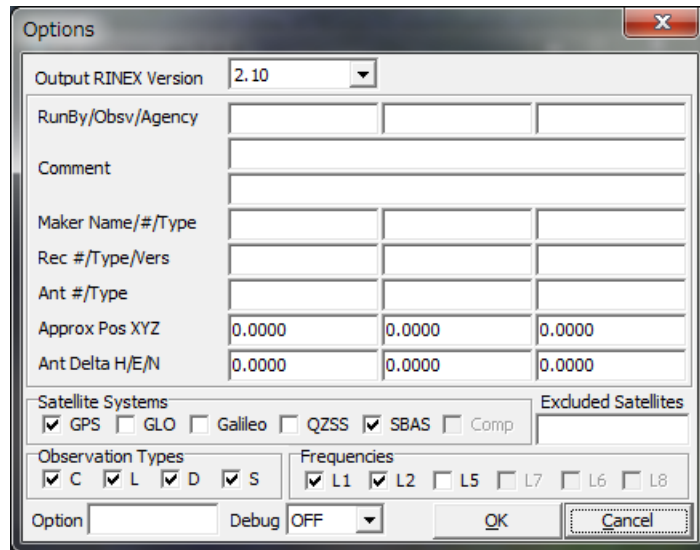
- (f) Superstar II : NovAtel Superstar II binary format
- (g) Hemisphere : Hemisphere Crescent/Eclipse binary format
- (h) SkyTraq : SkyTraq S1315F binary format
- (i) RINEX : RINEX OBS/NAV/GNAV/HNAV format.


If the wild-card (\*) is used in the file path, wild-card expanded multiple files 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.

- (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 : **.rtcm2**
- (b) RTCM3 : **.rtcm3**
- (c) NovAtel OEM4/V : **.gps**
- (d) u-blox : **.ubx**
- (e) Superstar II : **.log**
- (f) Hemisphere : **.bin**
- (g) SkyTraq : **.stq**
- (h) RINEX : **.obs, \*.o, \*.O**

- (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) 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 and HNAV are supported only in RINEX version 2. If selecting version 3 as output RINEX, all navigation data are outputted to a NAV type RINEX file.
- (5) You can set the start time or end time optionally by checking and setting 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.
- (6) You can push  button to configure RINEX options. Set the options for RINEX headers, navigation systems, observation types or frequencies with "Options" dialog .

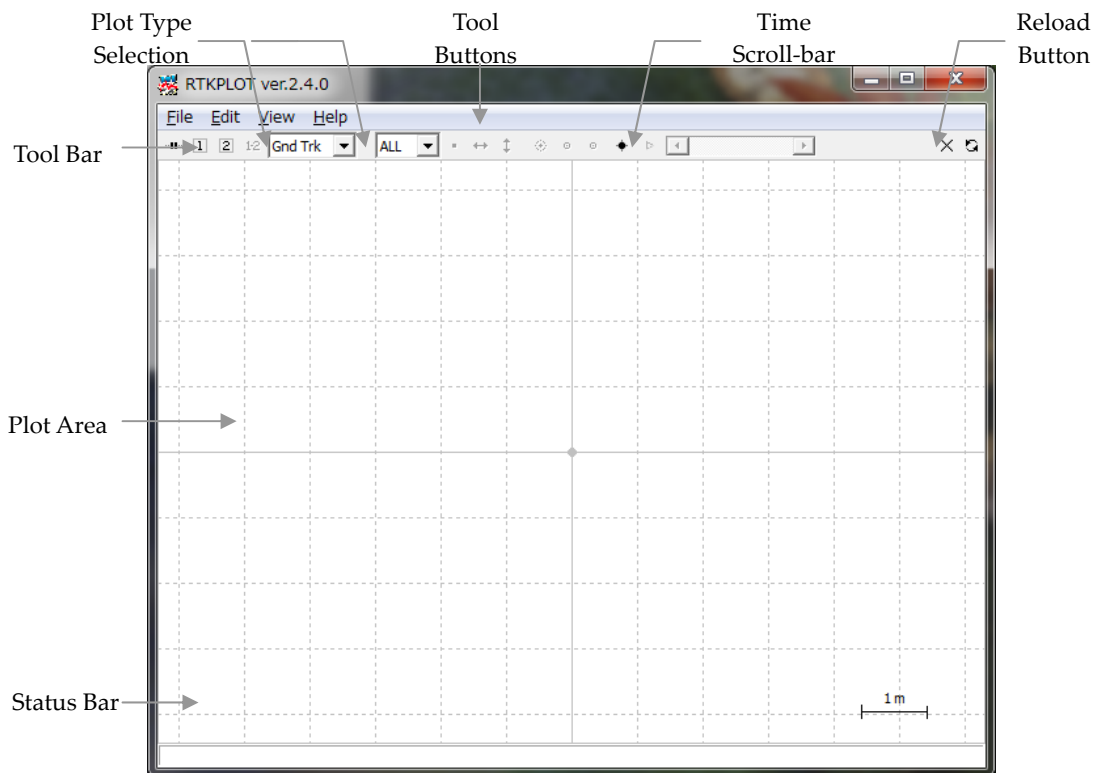


- (7) Push **Convert** button to start converting the receiver raw data to RINEX and SBAS log files. If you want to stop the conversion on the way, push **Abort** button. The status is displayed in the message area lower center in the main window. The message **O=nnn** means the number of converted observation data (epochs). The message **N=nn**, **G=nn**, **H=nn**, **S=nnn** and **E=nnn** means the number of navigation messages (NAV, GNAV and HNAV), SBAS messages and errors, respectively.
- (13) 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  button.

### 3.7 View and Plot Solutions and Observation Data with RTKPLOT

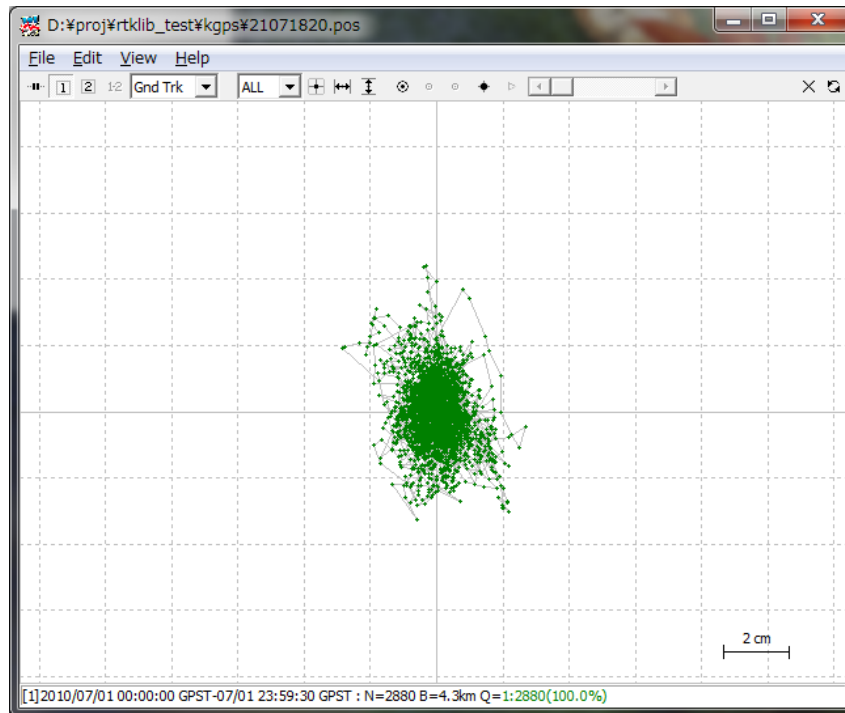
RTKLIB contains the AP RTKPLOT to view and plot the positioning solutions by RTKLIB or RINEX observation data with graphical user interface.

- (1) Execute the binary AP file `rtklib_<ver>\bin\rtkplot.exe`. You can see the main window of RTKPLOT. By pushing `Plot...` button or some buttons of RTKPOST and RTKCONV, RTKPLOT is also executed.

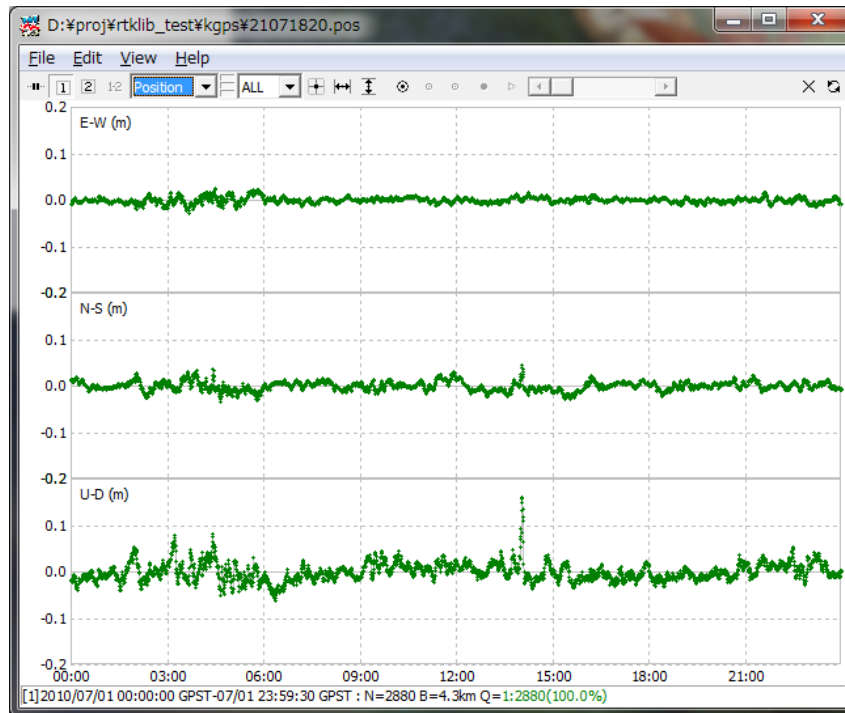


- (2) Execute the menu "File" - "Open Solution 1" and select the solution file with the file selection dialog. The input solution file can be RTKLIB solution format or NMEA-0183. If the file format is NMEA, the file has to contain at least GPRMC and GPSSGA sentences. If the solution file is valid, the 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 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

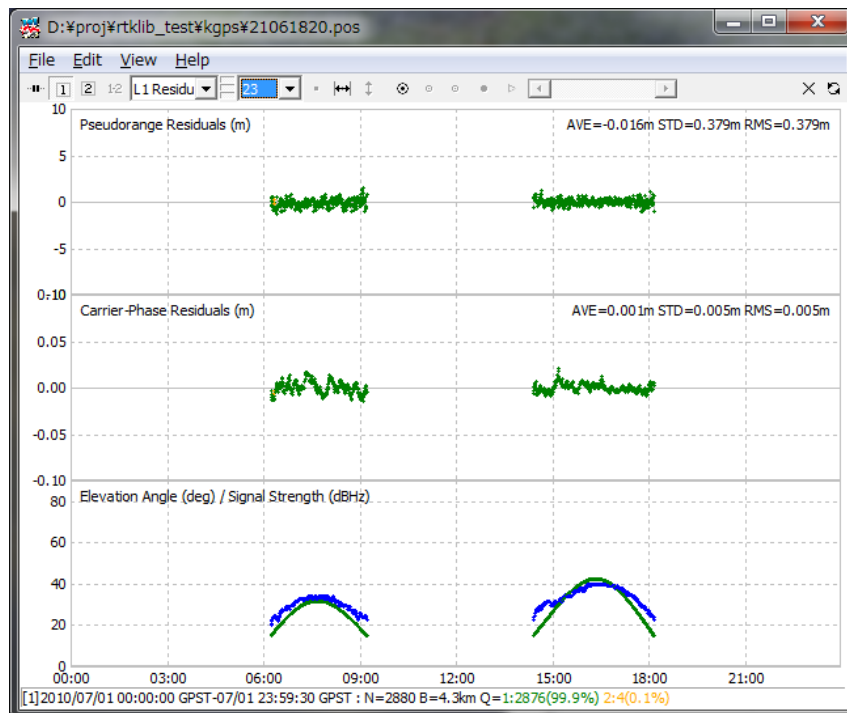
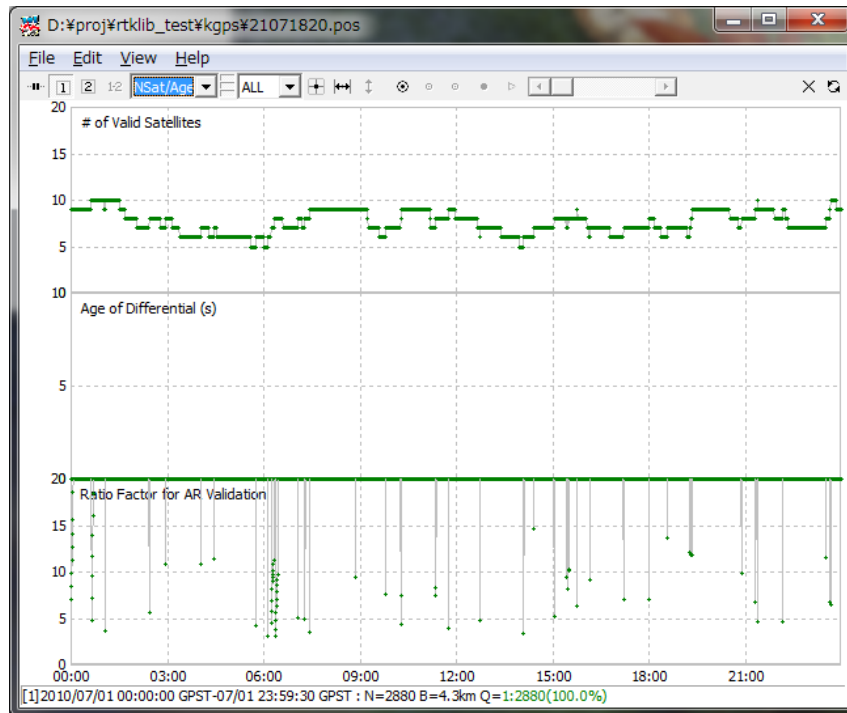
plot the solution file.













- (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 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 the left-button dragging and change the scale with the right-button dragging at the X/Y-axis area.





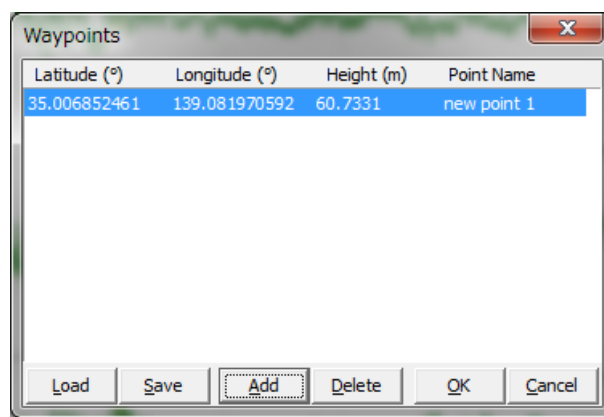
- (5) By selecting the pull down menu right in Tool Bar, you can switch the plot to NSat/Age/Ratio (number of valid satellites, age of differential, ratio factor of ambiguity validation). If you set Output Solution Status option to Residuals, you can show the residuals plot by selecting L1 Residuals or L2 Residuals. In residuals plot mode, you can select a satellite with right pull down menu as well as all satellites. In the residuals plot of carrier-phase, 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).






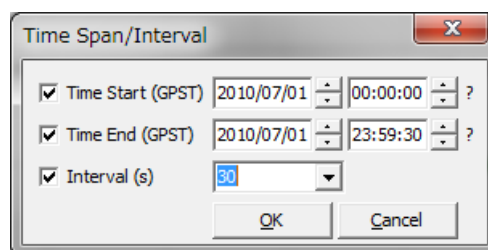
- (6) By operating tool buttons in Tool Bar, you can center the current position with , adjust the scale of X-axis with , adjust the scale of Y-axis with , display the current position as a large mark with , fix the current position at the horizontal center with , fix the current position at the vertical center with , start animation with  and stop animation with . You can also slide Time Scroll Bar to

change the epoch of the current position. To clear read data, execute the menu "File" - "Clear" or push  button in Tool Bar. To reload the solution file, execute the menu "File" - "Reload" or push  button in Tool Bar.

- (7) By executing the menu "View" - "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  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.



- (8) 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   buttons in Tool Bar. To Plot the difference of the solution 1 and the solution 2, push  button in Tool Bar.
- (9) 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.




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

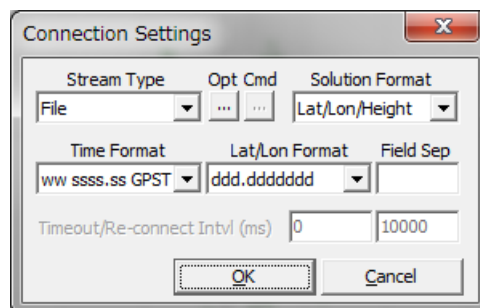
```


program : RTKPOST ver.2.4.0
inp file : D:\proj\rtklib_test\kgps\21061820.10o
inp file : D:\proj\rtklib_test\kgps\21071820.10o
inp file : D:\proj\rtklib_test\kgps\21071820.10n
obs start : 2010/07/01 00:00:00.0 GPST (week1590 345600.0s)
obs end : 2010/07/01 23:59:30.0 GPST (week1590 431970.0s)
pos mode : kinematic
freqs : L1+L2
solution : forward
elev mask : 15.0 deg
snr mask : 0.0 dBHz
dynamics : off
tidecorr : off
ionos opt : broadcast
tropo opt : saastamoinen
ephemeris : broadcast
amb res : fix and hold
val thres : 3.0
antenna1 : TRM29659.00 NONE ( 0.0000 0.0000 0.0000)
antenna2 : TRM29659.00 NONE ( 0.0000 0.0000 0.0000)
ref pos : 34.971863857 139.101688195 54.8876

(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
2010/07/01 00:00:00.000 35.006852461 139.081970592 60.7331 1 9 0.0057 0.0

```

- (11) To plot solutions in real-time, execute the menu "File"->"Connect 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. After setting the connection parameters, execute the menu "File"->"Connect" or push  button on 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.

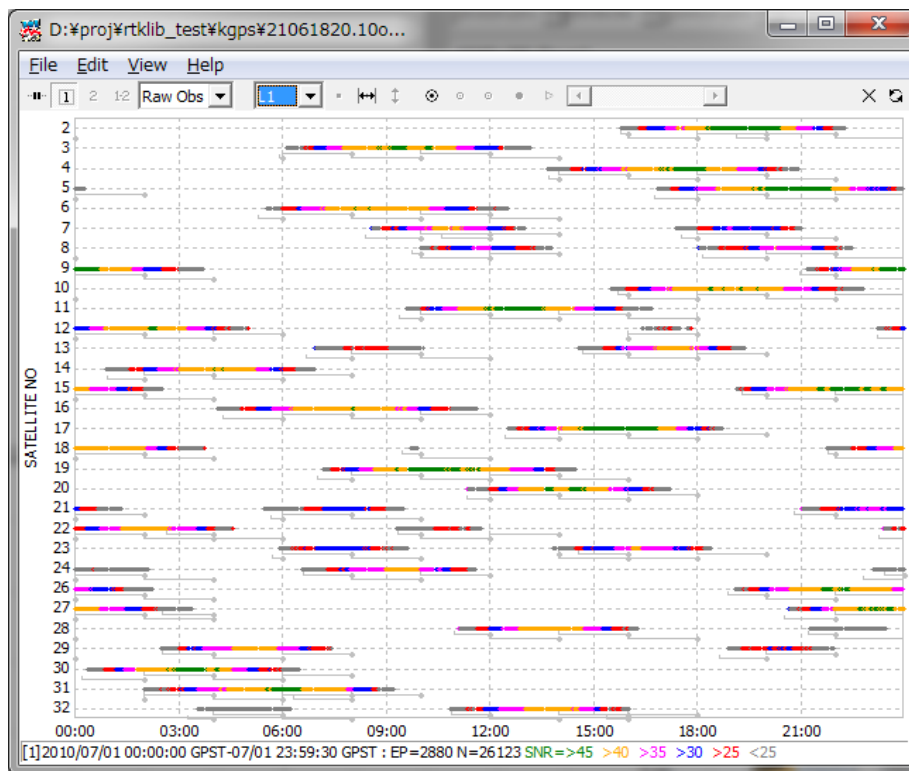


- (12) By pushing  button of RTKNAVI, RTKPLOT automatically runs and connects to the monitor port of RTKNAVI. If you want to connect to RTKNAVI executed on the remote PC, input the connection setting like TCP Client as Stream Type, IP address of the remote PC and port number of monitor port of RTKNAVI and connect to the remote RTKNAVI. In this case, multiple client

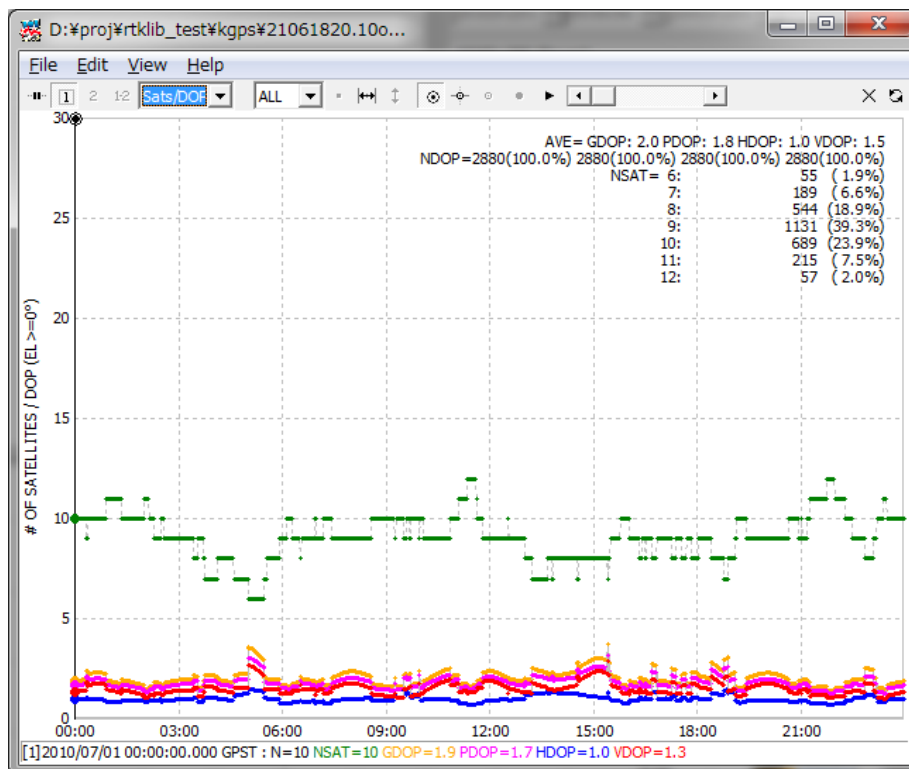
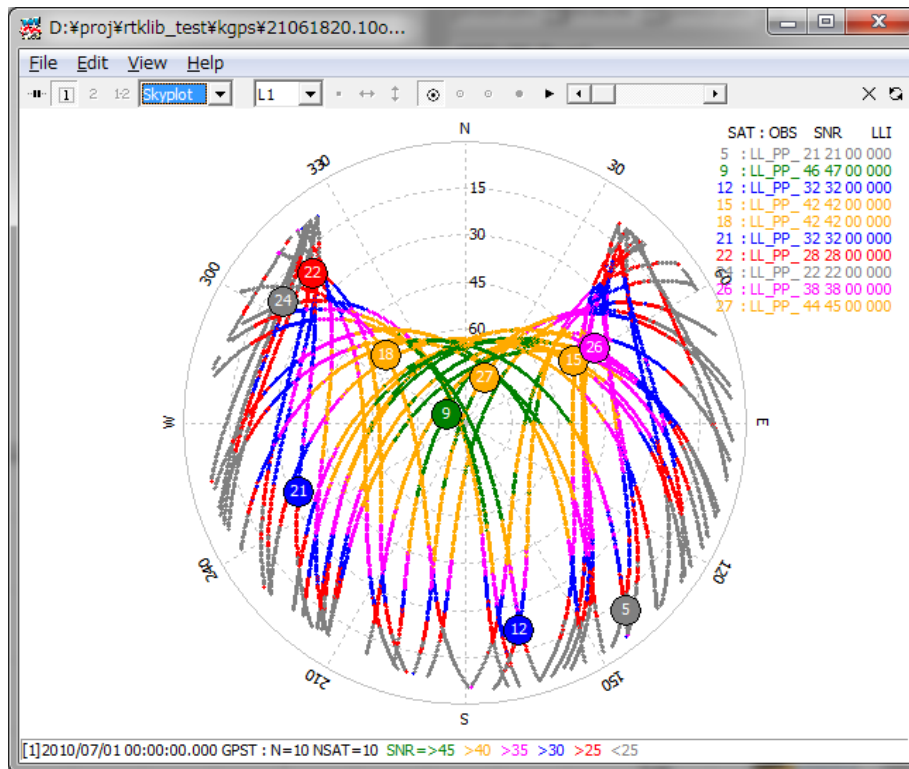


connections are allowed.

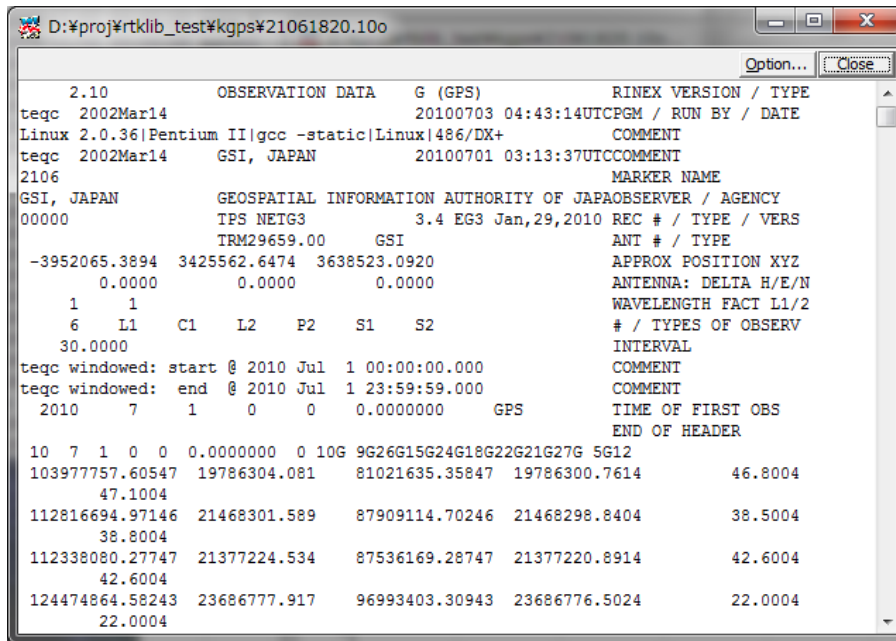
- (13) To plot the RINEX observation data, execute menu "File" -- "Open Obs Data" and select the RINEX observation and navigation message files. You can select multiple RINEX files. If you do not select the RINEX navigation message file, RTKPOST reads the observation data file paths with the extension replaced by `.nav` and `.gnav` (`.obs`) or `.yyn` and `.yyg` (`.yyo`) as the GPS and GLONASS 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 the satellite visibility or the observation data availability plot. You can use left-button or right-button drag to change the time span. You can also use some tool buttons in Tool Bar as well as for the solution plot.



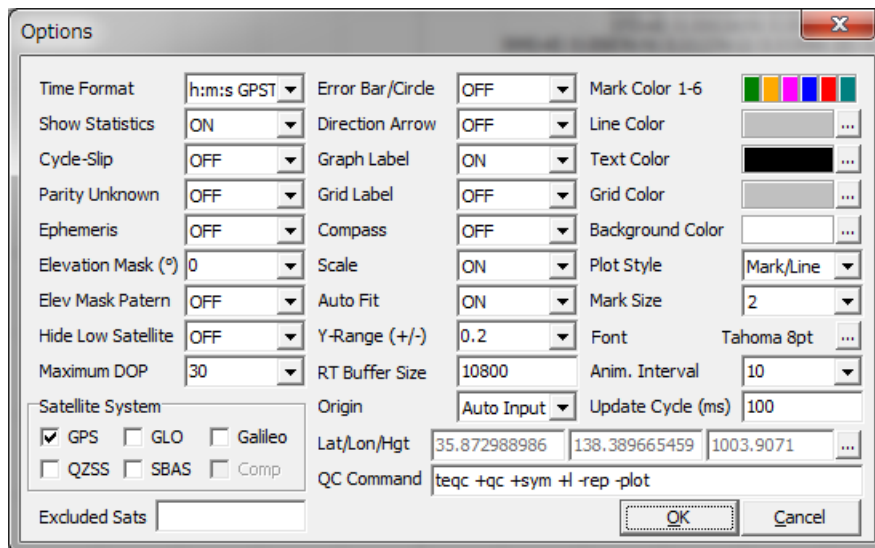
- (14) By selecting the pull down menu in Tool Bar, you can switch the plot to satellite visibility in skyplot (SkyPlot) or plot of number of visible satellites and DOP (Sats/DOP).



(15) 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.



(16) To configure the plotting options, execute menu "Edit" - "Options..." and set the options with the following "Options" dialog.



Item	Descriptions	Notes
Time Format	Select time format.	
Show Statistics	Set whether statistics are shown or not.	
Cycle-Slip	Set whether cycle-slip position is shown or not in observation data display. Cycle-slips are shown as red vertical lines in the observation data plot.	

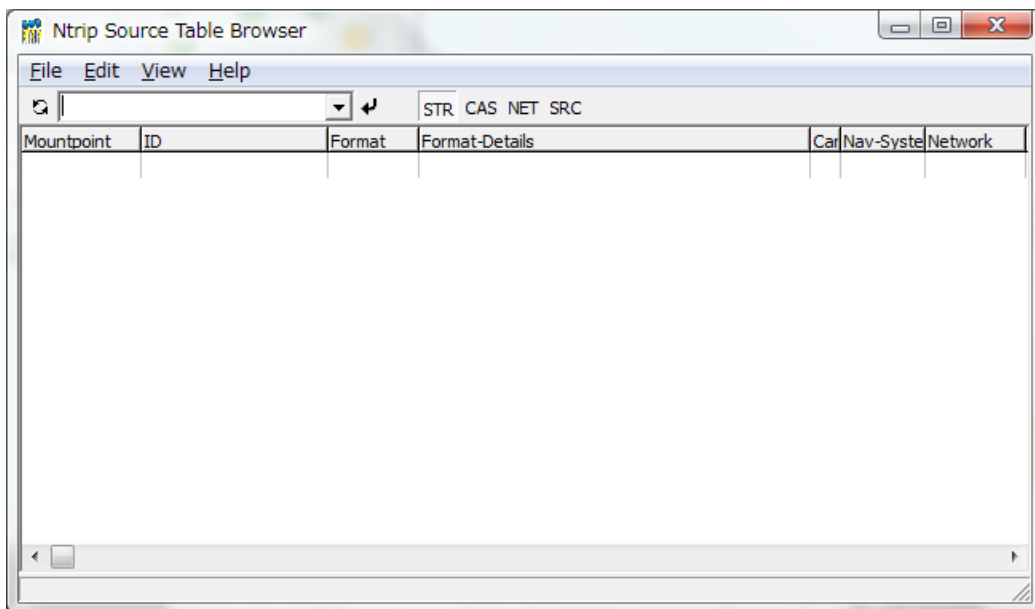
Item	Descriptions	Notes
Parity Unknown	Set whether parity unknown status is shown or not in observation data display. Parity unknown epochs are shown as gray vertical lines in the observation data plot.	
Ephemeris	Set whether ephemeris status is shown or not in observation data display. Ephemerides are shown as the grey line under the observation data plot. Gray dots means the Toe. Red ephemeris plot means the satellite is unhealthy.	
Elevation Mask	Set the elevation mask angle (deg) for observation data display. The elevation mask is also used for DOP computation.	
Hide Low Satellite	Set whether low elevation satellites under the elevation mask is shown or not.	
Maximum DOP	Set the y-axis limit of DOP plot.	
Satellite System	Select the navigation systems to be plotted.	
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 of Dots as the format.	
Direction Arrow	Set whether direction arrow and velocity arrow is shown or not in solution ground track display.	
Graph Label	Set whether graph labels are shown or not in solution display.	
Grid Label	Set whether grid labels are shown or not in solution display.	
Compass	Set whether compass is shown or not in solution ground track display.	
Scale	Set whether scale is shown or not in solution ground track display.	
Auto Fit	Set whether scale is adjusted automatically or not.	
Y-Range (+/-)	Set the range of Y-axis in solution display.	
RT Buffer Size	Set the real-time plot buffer size in epochs. The older solutions over the buffer size is deleted in real-time plot.	
Origin	Select the origin position of solution display as follows. - Start Pos, End Pos, Average Pos, Linear Fit Pos - Base Station, Lat/Lon/Hgt, Auto Input, Waypoint 1-10. 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.	
Mark Color	Set the mark colors in plots. Click color panel right and select color with color selection dialog.	
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.	
Mark Size	Set the mark size in plots.	
Anim. Interval	Set the animation interval for solution or observation data display.	
Update Cycle (ms)	Set the plot update cycle time in ms for real-time plotting	




Item	Descriptions	Notes
Font	Select the font in plots. Push ... button and select the font with font selection dialog.	
Lat/Lon/Hgt	Set latitude, longitude and height for the origin. Fill in the values directly or push ... button and select the 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.	

### 3.8 NTRIP Source Table 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.



- (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.

The screenshot shows the 'Ntrip Source Table Browser' window. The address bar displays 'www.igs-ip.net:2101'. The main table lists various stations with their respective details. The columns are: Mountpoint, ID, Format, Format-Details, Call, Lat, System, Network, Count, Latitude, Longitude, W, Sol, Generator, Comd, Aut, Fee, and Bit. The table is sorted by the 'STR' column, as indicated by the 'STR' button being highlighted in the top navigation bar.

(4) By pushing field title, you can sort the list. 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.

The screenshot shows the 'Ntrip Source Table Browser' window with the 'source table received' content displayed. The content includes headers like 'SOURCETABLE 200 OK' and 'Server: NTRIP Caster 1.5.8/1.0'. Below this, there is a list of station details for 'www.igs-ip.net:2101', including the URL, server information, and a list of stations with their respective details (ID, Format, Format-Details, Call, Lat, System, Network, Count, Latitude, Longitude, W, Sol, Generator, Comd, Aut, Fee, Bit).

### 3.9 Use Console APs of RTKLIB

RTKLIB includes the following console APs. To execute the APs,

These APs only use standard C functions. So you can rebuild the APs on the non-Windows environment like LINUX, UNIX, MAC OS X and so on. RTKLIB also contains standard make environment for gcc. Refer Appendix A RTKLIB Console AP Commands for detailed the command line options for these APs.

(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.

The console AP SBSPOS and SBSDUMP are no longer supported in ver.2.4.0.



## 4 Develop and Link user AP with RTKLIB

RTKLIB provide the following general purpose C-functions callable from user application program (AP).

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 message handling

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 detailed library functions and source programs in RTKLIB.

## Appendix A Console AP Commands

### A.1 RTKRCV

#### SYNOPSIS

```
rtkrvc [-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 USR2 signal to the process. For configuration file, refer B.4.

#### OPTIONS

```
-s          start RTK server on program startup
-p port    port number for telnet console
-m port    port number for monitor stream
-d dev     terminal device for console
-o file    configuration file
-r level   output solution status file (0:off,1:states,2:residuals)
-t level   debug trace level (0:off,1-5:on)
```

#### 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.

`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. With out option val, prompt message is shown to input the value. The change of the processing option is not enabled 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 configuration file, refer B.4.

### OPTIONS

```
-?          print help
-k file     input options from configuration file [off]
-o output   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]
-n      output NMEA-0183 GGA sentence [latitude/longitude/height]
-g      output latitude/longitude in the form of ddd mm ss.ss' [ddd.ddd]
-z      output height in geodetic height [ellipsoidal]
-t      output time in the form of yyyy/mm/dd hh:mm:ss.ss [sssss.ss]
-u      output time in utc [gpst]
-d col  columns of time under decimal point [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]
```

## A.4 CONVBIN

### SYNOPSIS

```
convbin [-ts y/m/d h:m:s] [-te y/m/d h:m:s] [-ti tint] [-r format] [-f freq]
        [-v ver] [-od] [-os] [-x sat] ... [-d dir] [-o ofile] [-n nfile]
        [-g gfile] [-h hfile] [-s sfile] file
```

### DESCRIPTION

Convert RTCM, receiver raw data log and RINEX file to RINEX and SBAS message file. SBAS message file complies with RTKLIB SBAS 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
NovAtel OEMV/4,OEMStar: RANGECPB, RANGE, RAWEPHEMB, IONUTCB, RAWWASSFRAMEB
NovAtel OEM3     : RGE, REGD, REPB, FRMB, IONB, UTCB
u-blox LEA-4T/LEA-5T : RXM-RAW, RXM-SFRB
NovAtel Superstar II : ID#20, ID#21, ID#22, ID#23, ID#67
Hemisphere      : BIN76, BIN80, BIN94, BIN95, BIN96
SkyTraq S1315F   : msg0xDD, msg0xE0, msg0xDC
RINEX           : OBS, NAV, GNAV, HNAV
```

### 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]
-r format       log format type
                rtc2= RTCM 2
                rtc3= RTCM 3
                nov  = NovAtel OEMV/4
                oem3 = NovAtel OEM3
                ubx  = ublox AEK-4T(LEA-4T/LEA-5T)
                ss2  = NovAtel Superstar II
```



```

hemis= Hemisphere Crescent/Eclipse
stq  = SkyTraq S1315F
rinex = RINEX
-f freq  number of frequencies [2]
-od      include doppler frequency [off]
-os      include snr [off]
-x sat   exclude satellite (multiple options are ok)
-d dir   output directory [same as input file]
-o ofile output RINEX OBS file   [<file>.obs]
-n nfile output RINEX NAV file   [<file>.nav]
-g gfile output RINEX GNAV file  [<file>.gnav]
-h hfile output RINEX HNAV file  [<file>.hnav]
-s sfile output SBAS message file [<file>.sbs]

```

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
*.ubx    u-blox AEK-4T(LEA-4T)
*.log    NovAtel Superstar II
*.bin    Hemisphere Crescent/Eclipse
*.stq    SkyTraq S1315F
*.nav,*.gnav,*.hnav,*.n,*.g,*.h" RINEX NAV/GNAV/HNAV

```

## 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 `ctr-c` in console if run foreground or send signal `SIGINT` for background process.

### OPTIONS

```
-in stream    input stream path
-out stream   output stream path
              serial://port[:brate[:bsize[:parity[:stopb[:fctr]]]]]
              tcpsvr://:port
              tcpcli://addr[:port]
              ntrip://[user[:passwd]@]addr[:port][:/mntpnt]
              ntrips://[user[:passwd]@]addr[:port][:/mntpnt[:str]]
              file://path
-d msec      status display interval (ms) [5000]
-s msec      timeout time (ms) [10000]
-r msec      reconnect interval (ms) [10000]
-n msec      NMEA request cycle (m) [0]
-c file      receiver commands file [no]
-p lat lon   NMEA position (latitude/longitude) (deg)
-t level     trace level [0]
-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	<p>Field indicator starting with "%" line follows after File header. To recognize the field formats, RTKLIB uses these lines. Do not delete them.</p>	
3	Solution body	<p>Solution body consists of the following fields. The field contents are varied according to the positioning options.</p>	
(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>WWW SSSSSS.SSS</b> : GPS week and TOW (time of week) in seconds.</p>	
(2)	Receiver Position	<p>The rover receive antenna or marker position estimated varied according to the positioning options.</p>	

No	Record/Field	Description	Notes
		<p><b>+ddd.dddddddd +ddd.dddddddd hhhh.hhhh :</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.</p> <p><b>+ddd mm ss.sss +ddd mm ss.sss hhhh.hhhh :</b> Latitude, longitude in degree, minute and second and height in m.</p> <p><b>+xxxxxxxxxx.xxxx +yyyyyyyyy.yyyy +zzzzzzzz.zzzz :</b> X/Y/Z components of ECEF frame in m.</p> <p><b>+eeeeeeee.eeee +nnnnnnnnn.nnnn +uuuuuuuu.uuuu :</b> E/N/U components of baseline vector in m. The local coordinate is referenced to the rover position.</p>	
(3)	Quality flag (Q)	<p>The flag which indicates the solution quality.</p> <p><b>1</b> : Fixed, solution by carrier-based relative positioning and the integer ambiguity is properly resolved.</p> <p><b>2</b> : Float, solution by carrier-based relative positioning but the integer ambiguity is not resolved.</p> <p><b>3</b> : Reserved</p> <p><b>4</b> : DGPS, solution by code-based DGPS solutions or single point positioning with SBAS corrections</p> <p><b>5</b> : Single, solution by single point positioning</p>	
(4)	Number of valid satellites (ns)	The number of valid satellites for solution estimation.	
(5)	Standard deviations (sdn, sde, sdu, sdne, sdeu, sdun)	<p>The estimated standard deviations of the solution assuming a priori error model and error parameters by the positioning options.</p> <p>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.</p>	
(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.	

**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 349007 137 25 : C666A0A7F1FE6000027F8000000346800000000000000000000000000000000000000
1471 349007 129 25 : C666A0A7F4FE6004047F8000000346800000000000000000000000000000000000000
1471 349008 129 4 : 53109FFFFF5FFDFFDFFDFFDFFC7FA9FFDFFDFFDFFDFFDFF9BBBBB33FFFFC0
1471 349008 137 4 : 53129FFC001FFDFFDFFDFFFA0009FFDFFDFFDFFDFFDFF9BBBBB33FFFFC0
1471 349009 137 3 : 9A0C9FFDFFDFFDFFDFFC017FF9FFDFFC009FFC015FFFB97B9BB9FBB80
1471 349009 129 2 : 9A0A9FFFC9FFFE9FFDFFDFFDFFDFFDFFDFFDFFDFF7F93FFBE79BBBBB9FA00
1471 349010 137 2 : C60A9FFFD1FFFDFFDFFDFFDFFDFFDFFDFFC003F88003E79FBBFB9FA00
1471 349010 129 3 : C60E9FFDFFDFFFE9FFC007FEDFFDFFDFFDFFDFFDFFDFFDFFB97B9BB9FBB80
1471 349011 137 26 : 536A0029E0EF0FF05F829C11C076033015A09D047037C1DE14F08FE000
1471 349011 129 26 : 536A0821A0DD05E82B813E0EF0F7897C27C12E08683B419C0BE057E000
1471 349012 137 28 : 9A723440E44E810029FF1F1F379C0BC35D4BE2B8078F15903253960200
1471 349012 129 28 : 9A722CB5D8739087B46B107DA8D9E828694B55F843782100AF146AD980
1471 349013 129 9 : C62434198D3F5D92BA855704800236DFE84FE06FFA47FE7FF0008E0240
1471 349013 137 9 : C6260C198D32310732404C1D40183CDFD187C8F3FF7FFD800FF7D6BE40
1471 349014 129 4 : 53119FFFF9FFDFFDFFDFFD3FA5FFDFFDFFDFFDFFDFF9BBBBB33FFFFC0
1471 349014 137 4 : 53109FFC005FFDFFDFFDFFFAFFDFFDFFDFFDFFDFFDFFDFF9BBBBB33FFFFC0
1471 349015 129 2 : 9A089FFFFC5FFFEDFFDFFDFFDFFDFFDFFDFFDFFDFFDFFDFF9BBBBB9FA00
1471 349015 137 3 : 9A0D9FFDFFDFFDFF9FFC017FFDFFDFFC00DFFC015FFFB97B9BB9FBB80
1471 349016 137 2 : C6089FFFD5FFC001FFDFFDFFDFFDFFDFFDFFDFFDFFDFFDFF9BBBBB9FA00
1471 349016 129 3 : C60C9FFDFFDFFFE5FFC007FF1FFDFFC001FFFDFFDFFB97B9BB9FBB80
1471 349017 137 25 : 5366587803FE3FF0010080FFF835E8000000000000000000000000000000000000
1471 349017 129 25 : 5366587FFDFEDFF4040080000035E80000000000000000000000000000000000000
1471 349018 129 63 : 9AFC0000000000000000000000000000000000000000000000000000000000000000000000
1471 349018 137 63 : 9AFC0000000000000000000000000000000000000000000000000000000000000000000000
1471 349019 129 26 : C66A0C53E26F09704781DC0DE06702FC19E1EF09F047821C0EF05FE000
1471 349019 137 26 : C66A0431E17E0A704741F80DC05E827815C0EF09F877C2FC15E096E000
1471 349020 137 4 : 53119FFC001FFDFFDFFDFFDFF9C001FFDFFDFFDFFDFFDFF9BBBBB33FFFFC0
1471 349020 129 4 : 53129FFFF5FFDFFDFFDFFDFFC3FA9FFDFFDFFDFFDFFDFF9BBBBB33FFFFC0

```

## 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> <p>\$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</p>	
2	Velocity/ Acceleration States	<p>Estimated rover velocity and acceleration in the filter. The format of a record is as follows.</p> <p>\$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<sup>2</sup>) float            velef/velnf/veluf : velocity e/n/u (m/s) fixed            accef/accnf/accuf : acceleration e/n/u (m/s<sup>2</sup>) fixed</p>	
3	Ionosphere Parameter States	<p>Estimated ionosphere parameter (vertical L1 ionosphere delay difference). The format of a record is as follows.</p> <p>\$ION,week,tow,stat,sat,az,el,ion,ion-fixed            week/tow : gps week no/time of week (s)            stat : solution status            sat : satellite id            az/el : azimuth/elevation angle(deg)            ion : vertical ionospheric delay L1 (m) float            ion-fixed: vertical ionospheric delay L1 (m) fixed</p>	
4	Troposphere Parameter States	<p>Estimated troposphere parameter (vertical troposphere delay residual). The format of a record is as follows.</p> <p>\$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</p>	

No	Record/Field	Description	Notes
		ztdf : zenith total delay (m) fixed	
5	Receiver H/W bias States	<p>Estimated GLONASS receiver H/W bias difference. The format of a record is as follows.</p> <p>\$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,...)                      bias : h/w bias coefficient (m/MHz) float                      biasf : h/w bias coefficient (m/MHz) fixed</p>	
6	Residuals	<p>Residuals of pseudorange and carrier-phase observables. The format of a record is as follows.</p> <p>\$\$SAT,week,tow,sat,frq,az,el,resp,resc,vsat,snr,fix,slip,lock,outc,\                      slipc,rejc                      week/tow : gps week no/time of week (s)                      sat/frq : satellite id/frequency (1:L1,2:L2,...)                      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:float,2:fixed,3:hold)                      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</p>	

**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
$$SAT,1557,432000.000,13,1,298.4,24.1,-0.6732,0.0003,1,42,1,1,1,0,1,0
$$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. 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.

### EXAMPLE

```
# rtkrcv options for rtk (2010/08/07, v.2.4.0)

console-passwd      =admin
console-timetype    =gpst      # (0:gpst,1:utc,2:jst,3:tow)
console-soltype     =dms       # (0:dms,1:deg,2:xyz,3:enu,4:pyl)
console-solflag     =2         # (0:off,1:std+2:age/ratio/ns)
inpstr1-type        =serial     #
(0:off,1:serial,2:file,3:tcpsvr,4:tcpcli,7:ntripcli,8:ftp,9:http)
inpstr2-type        =tcpcli     #
(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        =ttyACM0:115200:8:n:1:off
inpstr2-path        =192.168.0.2:2101
inpstr3-path        =
inpstr1-format      =oem4       #
(0:rtcm2,1:rtcm3,2:oem4,3:oem3,4:ubx,5:ss2,6:hemis,7:skytraq,8:sp3)
inpstr2-format      =oem4       #
(0:rtcm2,1:rtcm3,2:oem4,3:oem3,4:ubx,5:ss2,6:hemis,7:skytraq,8:sp3)
inpstr3-format      =rtcm3      #
(0:rtcm2,1:rtcm3,2:oem4,3:oem3,4:ubx,5:ss2,6:hemis,7:skytraq,8:sp3)
inpstr2-nmeareq     =off        # (0:off,1:latlon,2:single)
inpstr2-nmealat     =0          # (deg)
inpstr2-nmealon     =0          # (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        =sol1_%Y%m%d%h%M.pos
outstr2-path        =sol2_%Y%m%d%h%M.pos
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        =rov_%Y%m%d%h%M.log
logstr2-path        =ref_%Y%m%d%h%M.log
logstr3-path        =cor_%Y%m%d%h%M.log
misc-svrcycle       =10         # (ms)
misc-timeout        =30000      # (ms)
misc-reconnect      =30000      # (ms)
misc-nmeacycle      =5000       # (ms)
misc-buffsize       =32768      # (bytes)
misc-navmsgsel      =rover      # (0:all,1:rover,1:base,2:corr)
misc-startcmd       =./rtkstart.sh
misc-stopcmd        =./rtkshut.sh
file-cmdfile1       =../../../../data/oem4_raw_lhz.cmd
file-cmdfile2       =../../../../data/oem4_raw_lhz.cmd
file-cmdfile3       =
pos1-posmode        =kinematic #
```

```

(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-soltype       =forward    # (0:forward,1:backward,2:combined)
pos1-elmask        =15         # (deg)
pos1-snrmask       =0          # (dBHz)
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)
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-exclsats      =           # (prn ...)
pos1-navsys        =1          # (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      =0          # (deg)
pos2-aroutcnt      =5         #
pos2-arminfix      =10        #
pos2-slipthres     =0.05      # (m)
pos2-maxage        =30        # (s)
pos2-rejionno      =30        # (m)
pos2-niter         =1         #
pos2-baselen       =0         # (m)
pos2-basesig       =0         # (m)
out-solformat      =11h       # (0:11h,1:xyz,2:enu,3:nmea)
out-outhead        =on        # (0:off,1:on)
out-outopt         =off       # (0:off,1:on)
out-timesys        =gpst      # (0:gpst,1:utc,2:jst)
out-timeform       =tow       # (0:tow,1:hms)
out-timendec       =3         #
out-degform        =deg       # (0:deg,1:dms)
out-fieldsep       =          #
out-height         =ellipsoidal # (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      =0         # (s)
out-nmeaintv2      =0         # (s)
out-outstat        =off       # (0:off,1:state,2:residual)
stats-errratio     =100       #
stats-errphase     =0.003     # (m)
stats-errphasee1   =0.003     # (m)
stats-errphasee1l  =0         # (m/10km)
stats-errdoppler   =1         # (Hz)
stats-stdbias      =30        # (m)
stats-stdiono      =0.03      # (m)
stats-stdtrop      =0.3       # (m)
stats-prnaccelh    =1         # (m/s^2)
stats-prnaccelv    =0.1       # (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       =11h       # (0:11h,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant1-pos1          =0         # (deg|m)
ant1-pos2          =0         # (deg|m)
ant1-pos3          =0         # (m|m)
ant1-anttype       =          #
ant1-antdele       =0         # (m)
ant1-antdeln       =0         # (m)
ant1-antdelu       =0         # (m)
ant2-postype       =11h       # (0:11h,1:xyz,2:single,3:posfile,4:rinexhead,5:rtcm)
ant2-pos1          =35.87298031 # (deg|m)
ant2-pos2          =138.38966505 # (deg|m)
ant2-pos3          =1003.852   # (m|m)
ant2-anttype       =          #
ant2-antdele       =0         # (m)
ant2-antdeln       =0         # (m)
ant2-antdelu       =0         # (m)
misc-timeinterp    =off       # (0:off,1:on)
misc-sbasatsel     =0         # (0:all)
file-satantfile    =../../../../data/igs05.atx

```

```
file-rcvantfile =../../../../data/igs05.atx
file-staposfile =../../../../data/station.pos
file-geoidfile =
file-dcbfile =../../../../data/P1C1_ALL.DCB
file-tempdir =../../../../data/temp
file-geexefile =
file-solstatfile =
file-tracefile =
```

## Appendix C Library APIs (Application Program Interfaces)

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 `rtklib_<ver>\src`. The definition of data types regarding to the APIs, refer the header file `rtklib.h` in `rtklib_<ver>\src`.

RTKLIB API FUNCTION LIST ( \*: added in ver. 2.4.0, \*\*: modified in ver. 2.4.0)

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> *
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 vector	<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		
<code>str2num()</code>	String to number	<code>rtkcmn.c</code>
<code>str2time()</code>	String to time	<code>rtkcmn.c</code>
<code>time2str()</code>	Time to string	<code>rtkcmn.c</code>
<code>epoch2time()</code>	Calendar day/time to time	<code>rtkcmn.c</code>
<code>time2epoch()</code>	Time to calendar day/time	<code>rtkcmn.c</code>
<code>gpst2time()</code>	GPSTIME to time	<code>rtkcmn.c</code>
<code>time2gpst()</code>	Time to GPSTIME	<code>rtkcmn.c</code>
<code>gst2time()</code>	Galileo System Time to time	<code>rtkcmn.c</code> *
<code>time2gst()</code>	Time to Galileo System Time	<code>rtkcmn.c</code> *

Function	Description	Source Program
<code>time_str()</code>	Get Time String	<code>rtkcmn.c</code> *
<code>timeadd()</code>	Add time	<code>rtkcmn.c</code>
<code>timediff()</code>	Time difference	<code>rtkcmn.c</code>
<code>gpst2utc()</code>	GPS Time to UTC	<code>rtkcmn.c</code>
<code>utc2gpst()</code>	UTC to GPS Time	<code>rtkcmn.c</code>
<code>timeget()</code>	Get current time in UTC	<code>rtkcmn.c</code>
<code>timeset()</code>	Set current time in UTC	<code>rtkcmn.c</code> *
<code>time2doy()</code>	Time to Day of Year	<code>rtkcmn.c</code>
<code>adjgpsweek()</code>	Adjust GPS week number	<code>rtkcmn.c</code>
<code>tickget()</code>	Get current tick time	<code>rtkcmn.c</code>
<code>sleepms()</code>	Sleep for milliseconds	<code>rtkcmn.c</code>
<code>reppath()</code>	Replace file path	<code>rtkcmn.c</code> *
<code>reppaths()</code>	Replace file paths	<code>rtkcmn.c</code> *
	Coordinates functions	
<code>ecef2pos()</code>	ECEF to geodetic position	<code>rtkcmn.c</code>
<code>pos2ecef()</code>	Geodetic to ECEF position	<code>rtkcmn.c</code>
<code>ecef2enu()</code>	ECEF to local coordinates	<code>rtkcmn.c</code>
<code>enu2ecef()</code>	Local to ECEF coordinates	<code>rtkcmn.c</code>
<code>covenu()</code>	Covariance in local coordinates	<code>rtkcmn.c</code>
<code>covecef()</code>	Covariance in ECEF coordinates	<code>rtkcmn.c</code>
<code>xyz2enu()</code>	ECEF to ENU local coordinate transformation matrix	<code>rtkcmn.c</code>
<code>eci2ecef()</code>	ECI to ECEF transformation matrix	<code>preceph.c</code> *
<code>deg2dms()</code>	Convert degree to deg-min-sec	<code>rtkcmn.c</code> *
<code>dms2deg()</code>	Convert deg-min-sec to degree	<code>rtkcmn.c</code> *
	Input/output functions	
<code>readpos()</code>	Read station positions	<code>rtkcmn.c</code>
<code>sortobs()</code>	Sort observation data	<code>rtkcmn.c</code>
<code>screent()</code>	Screen data by time and interval	<code>rtkcmn.c</code>
<code>readnav()</code>	Read navigation data from file	<code>rtkcmn.c</code>
<code>savenav()</code>	Save navigation data to file	<code>rtkcmn.c</code>
	Debug trace functions	
<code>traceopen()</code>	Open trace file	<code>rtkcmn.c</code>
<code>traceclose()</code>	Close trace file	<code>rtkcmn.c</code>
<code>trace()</code>	Output trace	<code>rtkcmn.c</code>
<code>tracet()</code>	Output trace with time tag	<code>rtkcmn.c</code>
<code>tracemat()</code>	Output trace as matrix printing	<code>rtkcmn.c</code>
<code>traceobs()</code>	Output trace as observation data printing	<code>rtkcmn.c</code>
<code>traceonav()</code>	Output trace as GPS navigation messages	<code>rtkcmn.c</code>
<code>tracegnav()</code>	Output trace as GLONASS navigation messages	<code>rtkcmn.c</code>
<code>tracehnav()</code>	Output trace as GEO navigation messages	<code>rtkcmn.c</code> *
<code>tracepeph()</code>	Output trace as precise ephemeris	<code>rtkcmn.c</code>
<code>tracepclk()</code>	Output trace as precise clock	<code>rtkcmn.c</code> *
<code>traceb()</code>	Output trace as binary dump	<code>rtkcmn.c</code>
	Platform dependent functions	
<code>execcmd()</code>	Execute command	<code>rtkcmn.c</code>
<code>expath()</code>	Expand file path	<code>rtkcmn.c</code>
<code>createdir()</code>	Create directory	<code>rtkcmn.c</code> *

Function	Description	Source Program
Positioning models		
<code>satwavelen()</code>	Satellite signal carrier wave length	<code>rtkcmn.c</code>
<code>satazel()</code>	Satellite azimuth/elevation angle	<code>rtkcmn.c</code>
<code>geodist()</code>	Geometric distance	<code>rtkcmn.c</code>
<code>dops()</code>	Compute DOPs	<code>rtkcmn.c</code>
<code>csmooth()</code>	Carrier smoothing	<code>rtkcmn.c</code>
Atmosphere models		
<code>ionmodel()</code>	Ionospheric model	<code>rtkcmn.c</code>
<code>ionmapf()</code>	Ionospheric mapping function	<code>rtkcmn.c</code>
<code>tropmodel()</code>	Tropospheric model	<code>rtkcmn.c</code>
<code>tropmapf()</code>	Tropospheric mapping function (NMF)	<code>rtkcmn.c</code>
Antenna models		
<code>readpcv()</code>	Read antenna parameters	<code>rtkcmn.c **</code>
<code>searchpcv()</code>	Search antenna parameters	<code>rtkcmn.c *</code>
<code>antmodel()</code>	Receiver antenna model	<code>rtkcmn.c</code>
<code>antmodel_s()</code>	Satellite antenna model	<code>rtkcmn.c *</code>
Earth tides models		
<code>sunmoonpos()</code>	Sun and moon position	<code>preceph.c *</code>
<code>tidedisp()</code>	Displacements by earth tides	<code>ppp.c *</code>
Geoid model		
<code>opengeoid()</code>	Open external geoid file	<code>geoid.c</code>
<code>closegeoid()</code>	Close external geoid file	<code>geoid.c</code>
<code>geoidh()</code>	Geoid height	<code>geoid.c</code>
Datum transformation		
<code>loaddatum_p()</code>	Load datum transformation parameter	<code>datum.c</code>
<code>tokyo2jgd()</code>	Tokyo datum to JGD2000 datum	<code>datum.c</code>
<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>setrnxcdepri()</code>	Set RINEX code priority	<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>outrnxnavb()</code>	Output RINEX NAV body	<code>rinex.c</code>
<code>outrnxgnavh()</code>	Output RINEX GNAV header	<code>rinex.c **</code>
<code>outrnxgnavb()</code>	Output RINEX GNAV body	<code>rinex.c</code>
<code>outrnxhnavh()</code>	Output RINEX GEO NAV header	<code>rinex.c *</code>
<code>outrnxhnavb()</code>	Output RINEX GEO NAV body	<code>rinex.c *</code>
<code>uncompress()</code>	Uncompress file	<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>

Function	Description	Source Program
<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>
	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>crc32()</code>	CRC32 parity	<code>rtkcmn.c</code>
<code>crc24q()</code>	CRC24Q parity	<code>rtkcmn.c</code>
<code>decode_word()</code>	Decode navigation data word	<code>rcvraw.c</code>
<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_oem4f()</code>	Input OEM4/V raw data from file	<code>rcv/novatel.c</code>
<code>input_stq()</code>	Input SkyTraq raw data from stream	<code>rcv/skytraq.c</code>
<code>input_oem3f()</code>	Input OEM3 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>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>
	RTCM functions	
<code>init_rtcn()</code>	Initialize RTCM control	<code>rtcm.c</code>
<code>free_rtcn()</code>	Free RTCM control	<code>rtcm.c</code>
<code>input_rtcn2()</code>	Input RTCM2 message from stream	<code>rtcm.c</code>
<code>input_rtcn3()</code>	Input RTCM3 message from stream	<code>rtcm.c</code>
<code>input_rtcn2f()</code>	Input RTCM2 message from file	<code>rtcm.c</code>
<code>input_rtcn3f()</code>	Input RTCM3 message from file	<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>getsol()</code>	Get solution data from solution buffer	<code>solution.c *</code>

Function	Description	Source Program
<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>
<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>outsollexs()</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>outsollex()</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 sentence	<code>solution.c **</code>
<code>outnmea_gsv()</code>	Output NMEA GPGSV sentence	<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>sbstropcorr()</code>	SBAS tropospheric correction	<code>sbas.c *</code>
<code>sbsioncorr()</code>	SBAS ionospheric 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>



Function	Description	Source Program
<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>strgettime()</code>	Get current time from stream	<code>stream.c</code>
<code>strsendnmea()</code>	Send NMEA message to stream	<code>stream.c</code>
<code>strsendcmd()</code>	Send receiver command to stream	<code>stream.c</code>
<code>strsettimeout()</code>	Set stream timeout parameters	<code>stream.c *</code>
<code>strsetdir()</code>	Set local directory	<code>stream.c *</code>
	Integer ambiguity resolution	
<code>lambda()</code>	LAMBDA/MLAMBDA integer least-square estimation	<code>lambda.c</code>
	Standard positioning	
<code>pntpos()</code>	Standard positioning	<code>pntpos.c **</code>
	Precise positioning	
<code>rtkinit()</code>	Initialize RTK control struct	<code>rtkpos.c **</code>
<code>rtkfree()</code>	Free RTK control struct	<code>rtkpos.c</code>
<code>rtkpos()</code>	Precise positioning	<code>rtkpos.c **</code> <code>ppp.c</code>
<code>rtkopenstat()</code>	Open solution status file	<code>rtkpos.c</code>
<code>rtkclosestat()</code>	Close solution status file	<code>rtkpos.c</code>
	Post-processing positioning	
<code>postpos()</code>	Post-processing positioning	<code>postpos.c **</code>
	Stream server functions	
<code>strsvrinit()</code>	Initialize stream server	<code>streamsvr.c</code>
<code>strsvrstart()</code>	Start stream server	<code>streamsvr.c</code>
<code>strsvrstop()</code>	Stop stream server	<code>streamsvr.c</code>
<code>strsvrstat()</code>	Get stream server status	<code>streamsvr.c</code>
	RTK server functions	
<code>rtksvrinit()</code>	Initialize RTK server	<code>rtksvr.c</code>
<code>rtksvrstart()</code>	Start RTK server	<code>rtksvr.c **</code>
<code>rtksvrstop()</code>	Stop RTK server	<code>rtksvr.c</code>
<code>rtksvropenstr()</code>	Open output/log stream	<code>rtksvr.c *</code>
<code>rtksvrclosestr()</code>	Close output/log stream	<code>rtksvr.c *</code>
<code>rtksvrlock()</code>	Lock RTK server	<code>rtksvr.c</code>
<code>rtksvrunlock()</code>	Unlock RTK server	<code>rtksvr.c</code>
<code>rtksvostat()</code>	Get RTK observation data status	<code>rtksvr.c</code>
<code>rtksvrsstat()</code>	Get RTK stream status	<code>rtksvr.c</code>

The following APIs in ver. 2.3.0 are no longer available:

`uniqeph()`, `satposv()`, `satposiode()`, `postposopt()`, `pntvel()`, `eph2posp()`,  
`sbspntpos()`, `sbsupdatestat()`, `setsolopt()`, `setsolformat()`