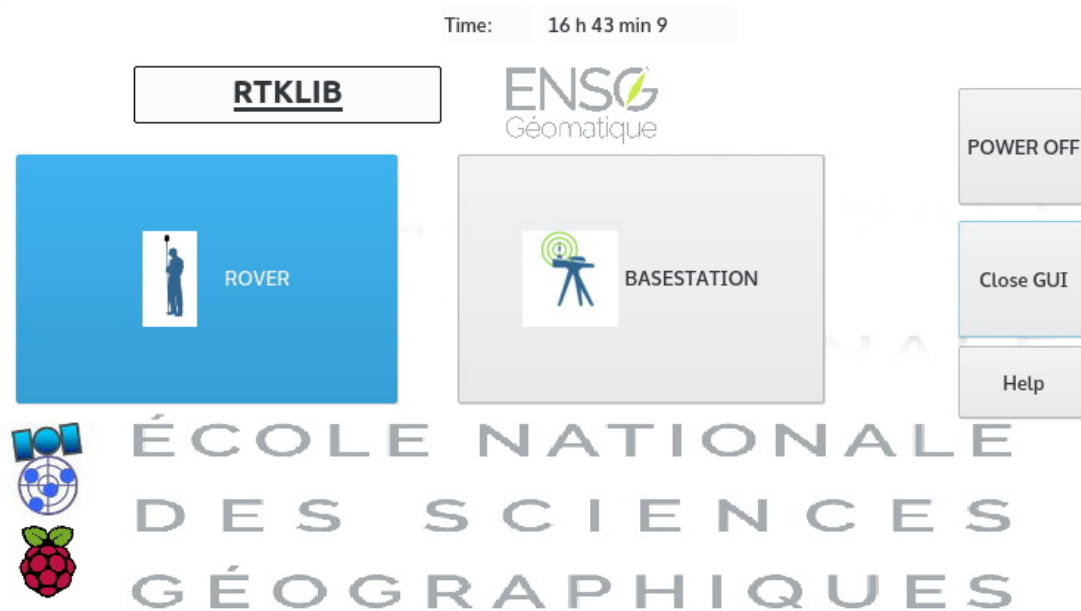


USER Manual

Touchscreen GUI for RTKLIB

II



SUMMARY:

Preliminary information:

I. Installation:

- 1.) Create the static library:
- 2.) and .pro project
- 3.) File access Rights

II. use

- 1.) Rover fashion:
- 2.) In Base mode:
- 3.) Configuration files:

Updated 22/05/2016

Preliminary information:

This project was requested by a "private" sponsor with the objective of achieving a positioning system in real-time at low cost. The system is based on RTKLIB. It is expected beforewhile for UNIX systems and formatted to best suit a Raspberry Pi2 equipped with a touch screen. All calculation results are from RTKLIB functions and thus their accuracy is directly related. This GUI program is intended only to make them easily accessible for a user using a UNIX OS, for a Base/Rover System. The software is built around Rtkrcv and Str2str two applications "CUI" of RTKLIB. This software is initially built to run under Raspberry with a touchscreen, without a keyboard or mouse. You have to edit the UBX files to suit your configuration (RF link, Ntrip user and GNSS stream).

Note 1:

At this stage the needs of the sponsor have yet to be evaluated through testing "in charge" rover/base that it will perform. Returns will adapt better to the GUI "Proven" needs. The changes will focus on the data to display the data to save in the point file, and on the basis of the method configuration options menu.

Note 2:

It remains to operate the Base mode beyond the only radio connection (functional), and understand the intricacies of str2str (Rtklib). These subtleties are on the command line allowing the transfer of the position of the base to the rover via radio.

Note 3:

The version used here is RTKLIB 2.4.2.p11, it can be updated to the latest version, just place the latest sources files of Rtklib in the lib folder before recompiling in Qtcreator

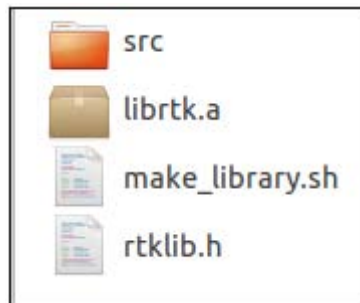
I. INSTALLATION

1.) Create the static library:

From a terminal (in the lib folder) start `./make-library.sh` order.

The file `librtk.a` appear if the construction of the library was carried out optimally.

The file `rtklib.h` remain at the same level as `librtk.a`.



2.) .pro files and project

Launch the `.pro` file in QT. Make a "clean all" then "run qmake" then "Build All" then "run". The RTKBASE interface must embark. Otherwise, go to 3.);

3.) File access Rights

Think about putting rights into "execution" of the `.pro` well as executable file.

Warning: Raspberry does not maintain an updated clock! Copying new files compile may result in error because the Raspberry notes file change schedules in the future ...

Check the time, adjust, via GPS or internet (automatic)!

Conf files:

Some configuration files "type" are included in the source code folder. They are at used to launch the "Rover" mode.

Note that:

- To perform the "NTRIP caster" must use "static";
- For the calculation on phase L1 must be used "kinematic";
- For differential on the code use the "dgps" mode.

II. USE OF THE GUI

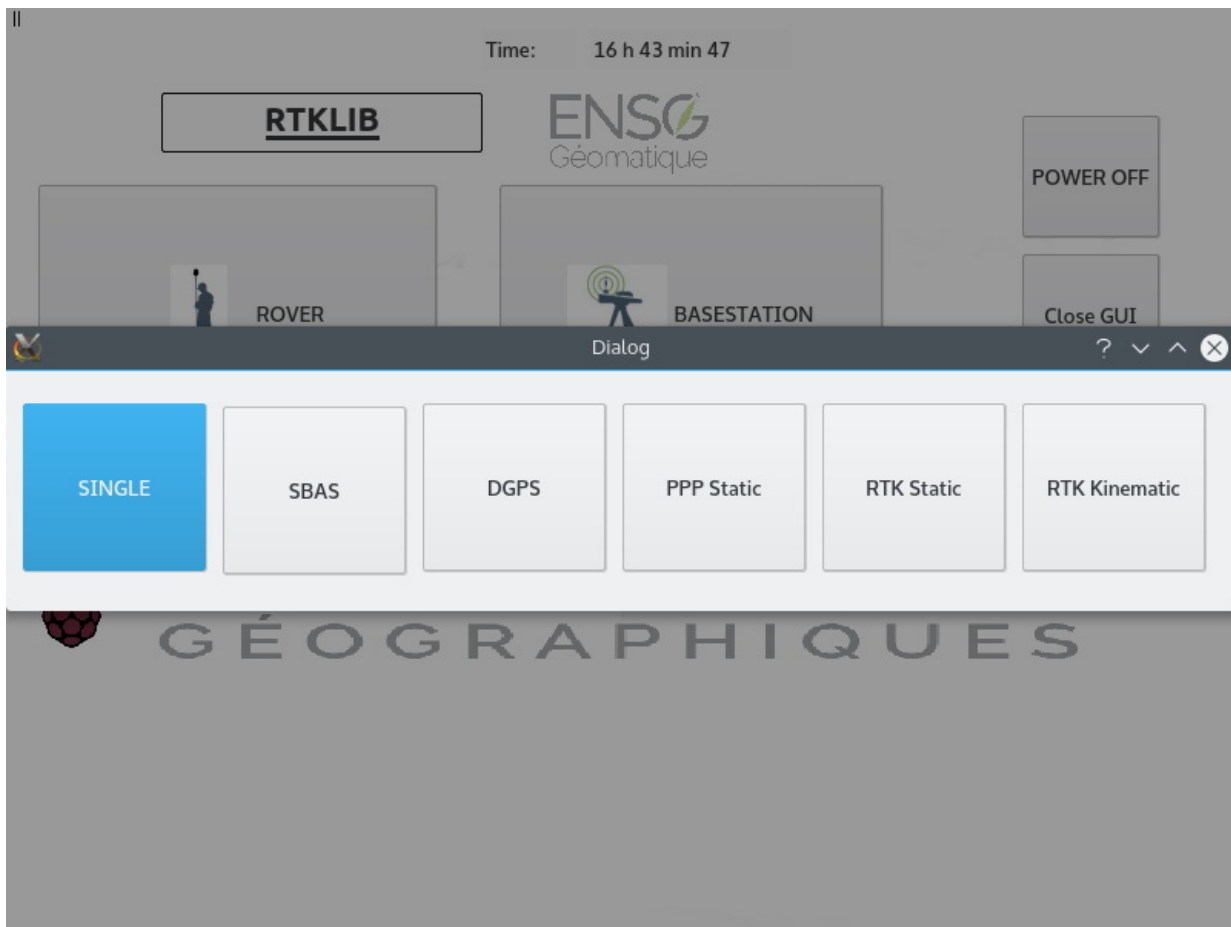
1.) Rover mode:

o Open the Rover mode;

o A window offering you different position calculation modes appears, choose one of the modes to start the position calculation, the main RTKlib calculation methods are available, to change these configurations you must manually edit * .ubx files in the folder /RTKBASE/conf/ files. In the statement of position it is advisable to choose the PPP-Static for better precision mode (If possible use a corrections via NTRIP)

II





6 modes are preset in the UBX files for Rover or calculating base position before using it as basestation:

- Single: RTKlib only calculates its position standalone without any correction
- SBAS: RTKlib uses the correction provided by the EGNOS satellites in Europe and the WAAS satellites in USA. the correction is provided by a ground stations and returned to the SBAS satellites that return them to GNSS receiver.
- DGPS: corrections RTKlib uses a ground station transmitted by RF link, GSM, or internet access, the proximity of correction stations provides a more locally adapted corrections than SBAS
- PPP-Static: RTKlib applies a smoothing on the path created by the raised positions to reduce positions errors (a kind of averaged) we can apply some DGPS corrections (to set corrections you must configure a Ntrip account access in the configuration files * .ubx
- RTK static: RTKlib performs a differential calculation between the base ideally located ideally in less than 70Km of the rover (the closer is the better). the distance and bearing between the rover and the base is taken into account (baseline). the calculation method knows that the rover is static.
- RTK kinematic: same calculation method as the RTK static but we know that the rover will move for the calculation. The resulting path is smoothed as for the PPP

If you want a USB key position log for post processing, configure log option in configuration files * .ubx The default configuration files are provided for Ublox, if you use a different GNSS you have to change the parameter input stream to the protocol of your constructor (NVS, Skytraq, etc)

o The window "Stream" opens. You will see activity at the Rover line.
 Otherwise there may be:
 GPS is not connected, or that your GPS is not communicating the right protocol (UBX, RTCM ...),
 or connection with the GPS is wrong.(USB/serial)
 If the problem is still there you can watch the log file to try to find the cause of the error, ...

Moreover, in RTK mode, information will also be posted on the received stream by the Rover from the base, or in correction mode, which is received from the Ntrip (Caster).

Stream								
Stream	Type	Fmt	S	In-byte	In-bps	Out-byte	Out-bps	Message
input rover	serial	ubx	C	33266	36289	0	0	
input base	tcpcli	oem4	C	0	0	0	0	connecting...
input corr	-	rtcm3	-	0	0	0	0	
output sol1	-	llh	-	0	0	0	0	
output sol2	-	nmea	-	0	0	0	0	
log rover	-	-	-	0	0	0	0	
log base	-	-	-	0	0	0	0	
log corr	-	-	-	0	0	0	0	
monitor	-	llh	-	0	0	0	0	

Example with a connexion error of the GNSS

Stream								
Stream	Type	Fmt	S	In-byte	In-bps	Out-byte	Out-bps	Message
input rover	serial	ubx	E	0	0	0	0	device open error (2)
input base	-	oem4	-	0	0	0	0	
input corr	-	rtcm3	-	0	0	0	0	
output sol1	-	llh	-	0	0	0	0	
output sol2	-	nmea	-	0	0	0	0	
log rover	-	-	-	0	0	0	0	
log base	-	-	-	0	0	0	0	
log corr	-	-	-	0	0	0	0	
monitor	-	llh	-	0	0	0	0	

To Choose the view you want, click on the window (anywhere except on buttons) and a new window will open. To access the options again later, just click anywhere on the window.



o Window "Status": Choose the type of units (llh or XYZ) for your solutions. The type of calculation mode selected (here DGPS) is displayed. The restitution level of the solution is shown in the field "status solution" means that the single solution is not the expected level.

 A screenshot of a software window titled "Status". At the top center is a "Status" label. To its right are two buttons for unit selection: "llh" (highlighted in orange) and "XYZ". Below these are several input fields organized into groups. On the left, there are labels for "Time to run:", "Pos Mode:", "Rover Single:", and "Rover Fixed:". The "Time to run:" field contains "00:01:09,2". The "Pos Mode:" field contains "DGPS". The "Rover Single:" group includes fields for "Lat = 48,796761 deg", "Lon = 2,130010 deg", and "H = 122,362510 m". The "Rover Fixed:" group includes fields for "X = 0,000000 m", "Y = 0,000000 m", and "Z = 0,000000 m". To the right of these are fields for "Solution status:" (containing "single"), "Sat Rover number: 6", "Sat Base number: 0", and "Valide Sat number : 4". At the bottom right is a "SAVE" button. A long empty text field is located at the bottom left.

To save the parked position, click "SAVE". The coordinates of the point (the point here) is saved in a file whose name format will be the type YYYYMMDD_HHMMSS.txt (eg 20141215_150322.txt). Files are stored in a folder "PointFiles" in the folder that runs the program. Date information and time correspond to the elements Rover. Hence the importance of putting in the time the Raspberry, failing to check the current time. The following items for the same session will also saved in the same file as a different number. If the rover mode is closed and opened again, a new file will be created.

Status

llh

XYZ

Time to run:

Pos Mode:

Solution status:

Rover Single:

Rover Fixed:

Point 0 saved in file

SAVE

o Window "Sat":

this view allows you to check the number and status of satellites observed. You can click on the "General" buttons, "elevation", "azimuth" in order to see the other view.

15%

Available satellites

General

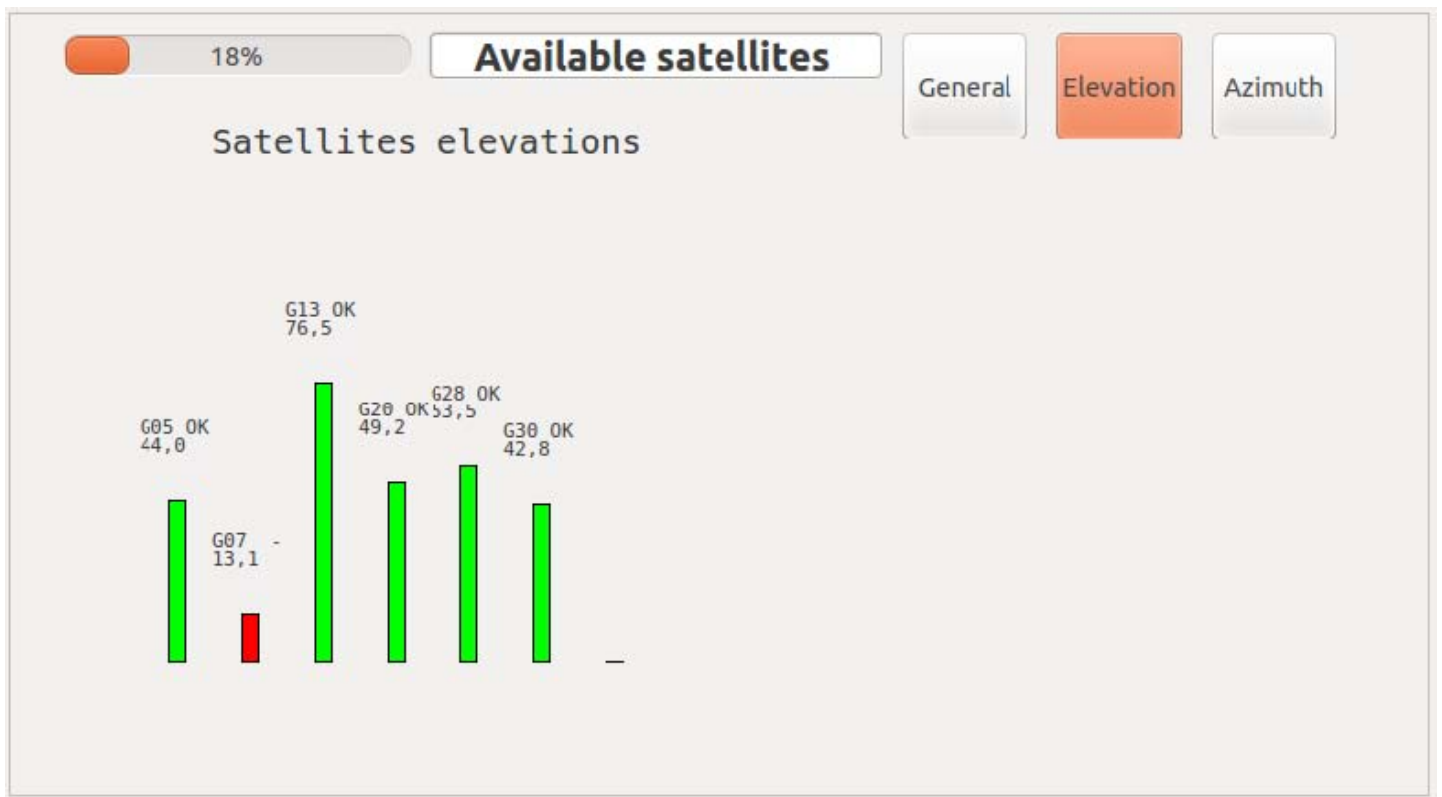
Elevation

Azimuth

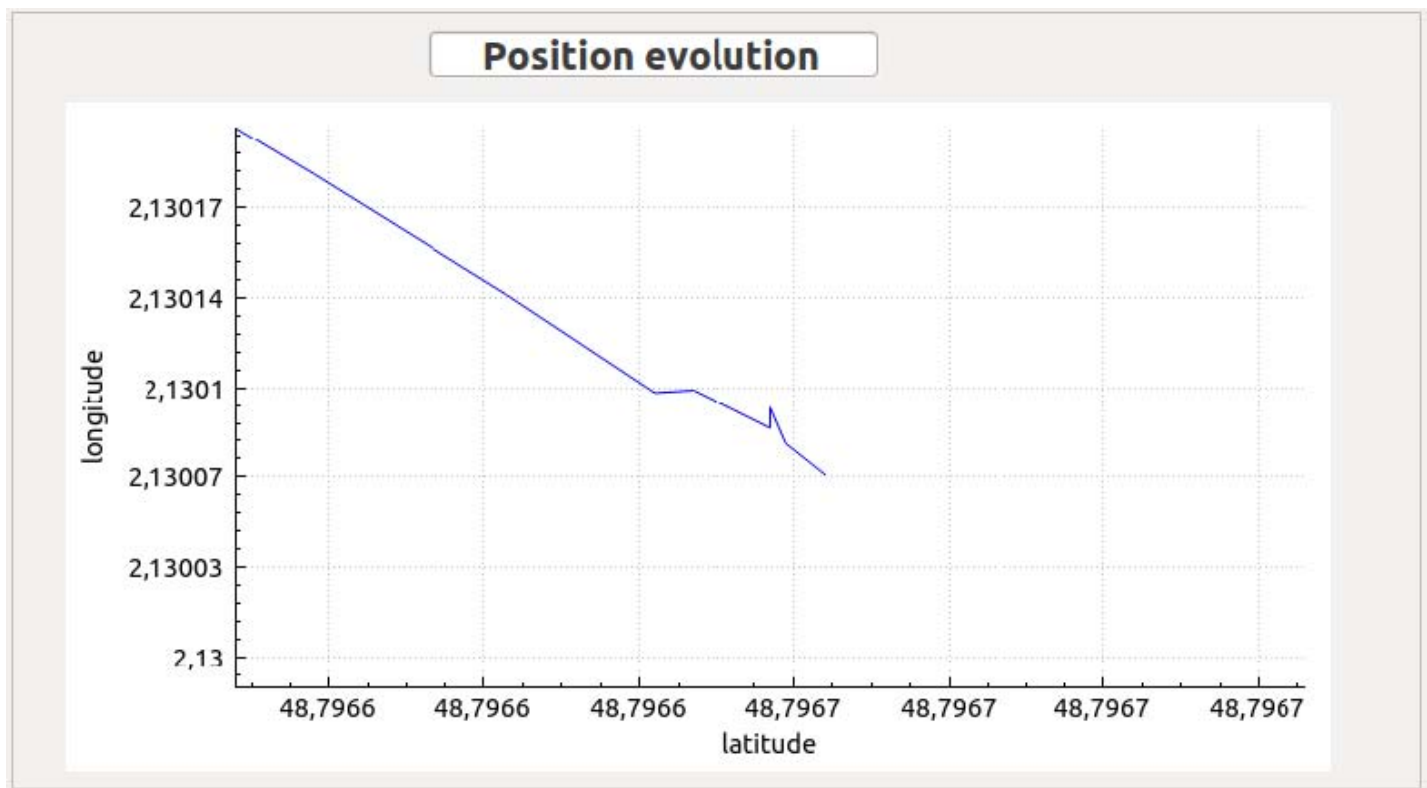
```

G05 OK, azimuth : 193,7, elevation : 44,2
G07 -, azimuth : 62,0, elevation : 13,3
G20 OK, azimuth : 275,0, elevation : 49,2
G28 OK, azimuth : 111,7, elevation : 53,5
G30 OK, azimuth : 61,2, elevation : 43,0

```

o Evolution Position 'window to see how the position changes over time. To date only the first 100 positions are stored by the Rover. We must improve this option by adapting the scale and duration of the monitoring points. This system is based on QCustomPlot, whose source file is integrated into the project.



o To close the mode, simply press the screen and in the options Press the button "CLOSE". You will return to the home page.

2.) In Base mode:

- o Open the "ROVER" mode. Let the unit until the position returned acceptable to you based on your need.
 - o Close the "ROVER" mode (the last position will be recorded) and then return to the main screen.
 - o Now select the "BASE" mode. Your radio-communication module must plugged and setup before. The following screen will then appear:
 - o Check the display of streams.
- If an error exists it will be reported to you.
- o Start the Base mode: the last position previously recorded will be recovered, the server will launch: check the stream display now.

The 'Dialog' window is used for configuring the in-stream and out-stream settings. It features two columns of radio buttons for selecting the stream type. The 'in stream' column includes options for RTCM2 (selected), RTCM3, Nov, Oem3, Ubx, Stq, Hemis, Javad, Nvs, Binex, and Ss2. The 'out stream' column includes the RTCM3 option. Below the radio buttons are two text input fields for command strings. The first field contains the command: `"-in","serial://ttyACM0:115200:8:n:1:#ubx","-msg","1004,1019,1012,1020,1006,1008"`. The second field contains the command: `"-in","serial://ttyACM0:115200:8:n:1:#ubx","-out","serial://ttyUSB0:57600:8:n:1:#RTCM3","-msg","1004,1019,1012,1020,1006,1008"`. To the right of each text field are buttons labeled 'default 1' and 'default 2'.

The 'Base Station Str2str status' window displays the current status of the base station. It includes a title bar with the text 'Base Station Str2str status'. Below the title bar is a large empty text area. The status information is displayed in a series of text boxes: 'Real Position picked up from rover mode', 'Lat = 48.796588 deg Lon = 2.130201 deg H= 140.953939 m', and '2016/01/23 15:41:45 [EC---] 20674 B 55584 bps'. A section titled 'CURRENT STR2STR OPTIONS' lists the following options: `-in`, `serial://ttyACM0:115200:8:n:1:#ubx`, `-msg`, and `1004,1019,1012,1020,1006,1008`. A 'CLOSE' button is located in the bottom right corner.

3.) Configuration files:

Some *.UBX configuration files are included in the /ConfFiles folder. They are used to launch the "Rover" mode. They can be edited manually in `texedit` to suit your configurations.

For more detail about setting the conf files you can look at the RTKLIB 2.42 manual

Note that :

- To perform the "NTRIP caster" must use "static";
- For the calculation on phase L1 must be used "kinematic";
- For differential on the code to "dgps".