PPP software package (version 1.4.3)

Table of Contents

1. Delivery package

The PPPWizard**xy** content delivery is composed following this schema below:

```
PPPWizardxy
|-- PPPSoftwarePackage_v1.4.3.pdf
|-- RTRover
|-- compile.sh
|-- generateLowLevel.cpp
|-- getStream.cpp
|-- laurichesse_ion_gnss_2015_september_bdp.pdf
|-- processLowLevel.cpp
|-- processStream.cpp
|-- rtklib
|-- tst
```

2. Content of the package

This package contains the following items:

- The "`rtklib`" directory contains the `rtklib` library source files modified to decode RTCM phase biases message (1265)
- The "`rtrover`" directory contains the PPP implementation (this is a library compatible with the `rtrover` interface of BNC)
- Sources files of executables (`getStream`, `processStream`, `processLowLevel`, `generateLowLevel`)
- The "`tst`"directory contains examples for each executable.
- Compilation and configuration files
- Documentation
- Feedbacks from users

3. Compilation of the package

3.1. COMPILATION UNDER LINUX

To compile both the libraries and executables, launch the '`./compile.sh`' command windows. This will create the rtklib and rtrover libraries, as well as the four executables getStream, processStream, processLowlevel and generateLowLevel.

3.2. COMPILATION UNDER MOBAXTERM (WINDOWS)

Following the described two steps to compile in MobaXterm

1. **Step 1**: install mobaxterm and the pluging `development.mxt3`
2. **Step 2**: include the following compilation directives: `CFLAGS="-O3 -lm -lpthread -lrt -static"` [Without the static flag, the executable program is generated but doesn't work due to an error in shared library]
3. **Step 3**: launch '`./compile.sh`' in a command window

3.3. COMPILATION UNDER MINGW (WINDOWS)

Proceed in the same way to compile under Windows environment

1. **Step 1**: Install the MinGW freeware with the following packages
   - `mingw-developer-tools`
   - `mingw32-base`
   - `mingw32-gcc-g++`
   - `msys-base`

2. **Step 2**: Edit PATH environment:
   - `<dir>\msys\1.0\bin`
   - `<dir>\bin`
   - `<dir>` being the MinGW installation directory

3. **Step 3**: Modify compilation options in the file RTRover\Makefile:

```
CFLAGS=-O3 -DRTRover_STATIC_LIB
```

4. **Step 4**: Modify compilation options in the file compile.sh :

```
CFLAGS="-O3 -Wl,--stack,4194304 -DRTRover_STATIC_LIB -lm -lWs2_32 -lWinmm"
```

5. **Step 5**: launch `sh compile.sh` in a command window

4. Description of the executables

From the 1.4 version, all output files carry out a new integrity indicator field (last column) describing the precise point integrity.

4.1. THE GETSTREAM EXECUTABLE

This executable uses the rtklib library to recover a set of streams from various sources and formats (local receiver, distant receiver (NTRIP), BRDC, SSR correction, etc…). These streams are then synchronized, assembled and converted in ASCII. The output file is self-sufficient and includes all the information needed for PPP. This file can then be processed directly (for example in a pipe) or stored for replay.

A typical run command is given by:

```
./getStream <conf_get.txt > stream.out
```

The configuration file of getStream is given from standard input and meets the following convention:

- stream address (rtklib convention)
- source type (rtklib convention)
- stream format (rtklib convention, for further decoding)

The receivers must be placed before the other streams in the configuration file.

The output file is written on the standard output.

Here is an example of the output:

```
1 1 1523285691 0D0A
2 1 1523285691 0D0A
3 1 1523285691 0D0A
1 1 1523285692 85473378F4883748B474F0EE7B1A68A4168CD60CC8B8BFF98D3D70D1F4FA9F77B5D37D8E27E0E777E4F78C46736CBF55CFF5
1 1 1523285692 4527586F0F0D6802D824042ACC03201E7E6D677AE3BDF6265E74B55E74C9647B8F14FCAC2A00F84480EC4573B617776FADF9
1 1 1523285692 DA0401473B889B7404F1E5058BA382E015FE22BF7DC05C012ACE80C7738054C6FF6BAF75775B824AF9F8FA74CD735CAB2ACA
```

To stop getStream, use the kill command with the PID of the getStream process.

4.2. THE PROCESSSTREAM EXECUTABLE

The processStream executable performs the PPP computation using the data recovered by getStream. It uses the following command:

```
./processStream –conf conf_process.txt –rover rover.txt –dcb "*.DCB" [-verbose] [-lowlevel]< stream.out
```

Where:

- conf_process.txt is the general configuration file (the configuration is the same for each rover)
- rover.txt contains the name of each receiver (in the same order of the getStream configuration file) and the rover a priori position (“0.0 0.0 0.0” if not used)
- “*.DCB” are the code biases files used by the PPP (P1C1, P1P2, P2C2). These files are available at ftp://ftp.aiub.unibe.ch/CODE/ and need to be changed once a month approximately.
- verbose : verbose mode (internal use only)
- lowlevel : generate measurements file (lowlevel.txt)

From the 1.4 version, the Ionospheric VTEC (Vertical Total Electron Content) is managed. To test it, the configuration is to set the 3rd column of `SigMesIono` to ‘`0.0`’ and then retrieve the same results as those given by the version 1.3.

4.2.1. THE CONFIGURATION FILE

The following table shows the structure of each entry of the general configuration file:

Parameter	Type/Unit	Comment	Typical value
mode	Enum	Processing mode : mode_PPP_DF, (PPP bi) mode_SPP_DF, (SPP bi) mode_PPP_SF, (PPP mono) mode_SPP_SF, (SPP mono) mode_PPP_AR, (PPP with ambiguity resolution)	mode_PPP_AR
antexFileName	String	ANTEX file name	
AR/JumpsIndicators	Boolean	AR/JumpsIndicators (N1/Nw/Ex)	1 1 0
useGPS	Boolean		1
useGlonass	Boolean		1
useGalileo	Boolean		1
useBeidou	Boolean		0
sbasCorrection	Boolean	0 -> RTIGS clock correction, 1 -> SBAS clock correction	0
convergence	Int/sec	Time between consecutive convergence (for convergence tests), 0 if no convergence	0
outputVerbose	Boolean	Verbose output	0
step	Real/sec	measurement interval	1
maxAge	Real/sec	maximum RTCM correction age	10
stepMin	Integer/S.U.	minimum step before AR	3600
maxReject	Integer/S.U.	RAIM maximum rejection	2
raim	Boolean	Advanced RAIM	1
mapThr	Real/S.U.	tropo mapping threshold (1/sin(ele))	6
sigIniTro	Real/m	tropo initial noise	0.5
sigModTro	Real/m	tropo model noise	0.000005
nbSatFixAmb	Integer/S.U.	minimum satellite for AR	0
threAmb	Real/m	ambiguity threshold for AR	0.01
sigIniBiasClk	Real/m	initial clock bias noise	0
sigModBiasClk	Real/m	model clock bias noise	0.001
sigIniIono	Real/m	initial iono noise	10
sigModIono	Real/m	model iono noise	0.002
sigMeasIono	Real/m	iono measurement noise	1
ionoThr	Real/m	iono measurement rejection threshold	5
sigMeasTropo	Real/m	tropo measurement noise	0.1
tropoThr	Real/m	tropo measurement rejection threshold	1
sigIniPos	Real/m	initial position noise	50 0 (position fixed)
sigModPos	Real/m	model position noise	10 (mobile receiver) 0.02 (static receiver) 0 (position fixed)
preDTMax	Real/sec	maximum measurement gap	300
codeThr	Real/m	code measurement rejection threshold	10
phaseThr	Real/m	phase measurement rejection threshold	0.05
sigMeasCodeGps	Real/m	code GPS measurement noise	1
sigMeasPhaseGps	Real/m	phase GPS measurement noise	0.01
sigMeasCodeGlo	Real/m	code Glonass measurement noise	5
sigMeasPhaseGlo	Real/m	phase Glonass measurement noise	0.01
sigMeasCodeGal	Real/m	code Galileo measurement noise	1
sigMeasPhaseGal	Real/m	phase Galileo measurement noise	0.01
sigMeasCodeBds	Real/m	code Beidou measurement noise (GEO/IGSO/MEO)	5 5 5
sigMeasPhaseBds	Real/m	phase Beidou measurement noise(GEO/IGSO/MEO)	0.01 0.01 0.01
Smooth	Real/S.U.	Doppler smoothing coefficient	0 of no smoothing 0.95 if smoothing

4.2.2. THE ROVER FILE

The rover file contains one line per rover. Each line has the following format:

Field	Comment
Name	Rover Name
X	Rover a priori position X ITRF 08
Y	Rover a priori position Y ITRF 08
Z	Rover a priori position Z ITRF 08

4.2.3. THE OUTPUT FORMAT

The output of processing consists in one line per epoch. Each line has the following format:

Field	Comment
Date	Calendar day *
Hour	Calendar hour *
Rover	Rover name
PPP mode	SPP, PPP, etc
Measurements	Total number of measurements
ExtraWidelanes	# measurements with fixed extra-widelanes
Widelanes	# measurements with fixed widelanes
Narrowlane	# measurements with fixed narrowlanes
X	X (m, ITRF 08)
CovX	X cov (m)
Y	Y (m, ITRF 08)
CovY	Y cov (m)
Z	Z (n, ITRF 08)
CovZ	Z cov (m)
DryTropo	Dry tropo (m)
Tropo	Estimated wet tropo (m)
Cov tropo	Wet tropo cov (m)
Integrity	Integrity indicator

If `outputVerbose` is activated in the configuration file, the output has the following format:

Field	Comment
Date	Calendar day *
Hour	Calendar hour *
Rover	Rover name
PPP mode	SPP, PPP, etc
HstaGPS	Clock GPS
HstaGlo	Clock Glo
HstaGal	Clock Gal
HstaBds	Clock Bds
Measurements	Total number of measurements
ExtraWidelanes	# measurements with fixed extra-widelanes
Widelanes	# measurements with fixed widelanes
Narrowlane	# measurements with fixed narrowlanes
X	X (m, ITRF 08)
CovX	X cov (m)
Y	Y (m, ITRF 08)
CovY	Y cov (m)
Z	Z (m, ITRF 08)
CovZ	Z cov (m)
DryTropo	Dry tropo (m)
Tropo	Estimated wet tropo (m)
Cov tropo	Wet tropo cov (m)
Integrity	Integrity indicator

4.2.4. THE VERBOSE OPTION

The verbose option of `processStream` ([-verbose]) generates the following output on the stderr. This output has one line per epoch. Each line has the following format:

Field	Comment
Rover	Rover number
Date	Calendar day
Hour	Calendar hour
Tropo	Tropo

And for each constellation and each satellite:

Field	Comment
PRN	For example : G01, R10, E11, C06
iono	iono

A specific RTCM message (99) has been created in order to use the verbose option output. This RTCM message is an internal message read by the "`decode_tropo_iono`" function.

4.2.5. THE LOWLEVEL FILE

The following table shows the structure of each line of the `lowlevel` file ([-lowlevel]):

Field	Title	Unit	Remarks
1	Rover Number	-	
2	Satellite	-	Example G01
3	CNES Julian Day	day	Day 1 at 01/01/1950
4	Seconds in the day	sec	
5	C1/P1/E1	m	Code
6	P2	m	
7	C6 (Beidou)	m	
8	C5/E5a (Galileo)	m	
9	E5b/C7 (Beidou)	m	
10	L1	cycle	Phase
11	L2	cycle	
12	L6 (Beidou)	cycle	
13	L5/L5a (Galileo)	cycle	
14	L5b/L7 (Beidou)	cycle	
15	D1	Hz	Doppler
16	D2	Hz	
17	D6 (Beidou)	Hz	
18	D5/D5a (Galileo)	Hz	
19	D5b/D7 (Beidou)	Hz	
20	Slot/typeBds		Slot Glo or type BDS: 0:GEO, 1:IGSO, 2:MEO
21	X	m	Satellite position APC at the emission time of the signal
22	Y	m	
23	Z	m	
24	H	m	Satellite clock at the emission time of the signal, including relativistic effect
25	Vx	m/s	Satellite velocity
26	Vy	m/s	
27	Vz	m/s	
28	Iono	m	Slant ionospheric delay At Band1 constellation frequency
29	Source Iono		1:precise, 2:SBAS, 3:Global
30	Tropo	m	Vertical ZTD

31	Yaw	circle	Satellite yaw angle
32	bC1	m	Code bias on Band1 Consistent with fields 5 and 24
33	bP2	m	
34	bC6	m	
35	bC5	m	
36	bE5b	m	
37	bL1	cycle	Phase bias on Band1 Consistent with fields 10 and 24
38	bL2	cycle	
39	bL6	cycle	
40	bL5	cycle	
41	bL5b	cycle	
42	N1 indicator	0/1	Biases compatible with N1 integer ambiguity
43	Wl indicator	0/1	Biases compatible with widelanes integer ambiguity
44	Discontinuity	-	Integer value for discontinuity

* The date is a multiple of the step.

4.3. THE FULL PPP SEQUENCE

Update your credentials in the file `conf_get.txt`.
To execute the entire PPP sequence (stream acquisition and processing) launch:

```
./getStream <conf_get.txt | ./processStream -conf conf_process.txt -rover rover.txt -dcb
"*.DCB"
```

Here is an example of the output:

```
20-06-16 20:59:01.000 GAMG PPP 26  0 12 12 -3191609.077 +- 0.009  4096901.204 +- 0.010  3691840.488 +- 0.009 2.104 + 0.047298 +- 0.000553
0.0015
20-06-16 20:59:02.000 GAMG PPP 26  0 12 12 -3191609.078 +- 0.009  4096901.203 +- 0.010  3691840.488 +- 0.009 2.104 + 0.047297 +- 0.000553
0.0014
20-06-16 20:59:03.000 GAMG PPP 26  0 12 12 -3191609.077 +- 0.009  4096901.203 +- 0.010  3691840.489 +- 0.009 2.104 + 0.047296 +- 0.000553
0.0013
20-06-16 20:59:04.000 GAMG PPP 26  0 12 12 -3191609.078 +- 0.009  4096901.203 +- 0.010  3691840.490 +- 0.009 2.104 + 0.047294 +- 0.000553
0.0013
```

4.4. THE GENERATELOWLEVEL EXECUTABLE

The Ionospheric SBAS interface is handled from the version 1.4. Further, a better management of the RF GLONASS Channel with warning (when file doesn't exist) is done.

The `generateLowLevel` executable can be used in order to create a `lowlevel` file from input files (dcb, atx…). It uses the following command:

```
./generateLowlevel -rinex rinex.rnx -sp3 ephem.sp3 -clk clock.clk -bias bias.bia -atx igs.atx
-dcb "*.DCB" -chan channelGlo.txt > lowlevel.txt
```

Where:

- `rinex.rnx`: measurement file
- `ephem.sp3`: ephemeris file
- `clock.clk`: clock file
- `bias.bia`: sinex bias file (if a GRG solution is processed, there is no need to specify this file for A.R.).
- `igs.atx` is the ANTEX file name
- "*.DCB" are the code biases files used by the PPP (P1C1, P1P2, P2C2). These files are available at ftp://ftp.aiub.unibe.ch/CODE/ and need to be changed once a month approximately.
- channelGlo.txt is a file containing channels for each Glonass satellite

See §4.2.5 for the description of the lowlevel file.

Here is an example of the output:

```
1 E03 25734       72000.000    28153049.555           0.000           0.000    28153051.067    28153049.236   147945272.290           0.000           0.000
110478616.540   113360658.784        2737.630           0.000           0.000        2044.458        2097.879 -999    16547210.584     7241744.546    23454342.077      -
93104.850       -1706.808        1616.238         706.674 0.000000 2 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0 1 0
1 E04 25734       72000.000    27781987.526           0.000           0.000    27781987.996    27781986.235   145995392.312           0.000           0.000
109022577.823   111866625.987       -3016.288           0.000           0.000       -2252.343       -2311.165 -999   -24442182.075    10841044.998   -12689448.056     -
164057.648        942.061        -857.924       -2544.729 0.000000 2 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0 1 0
1 E05 25734       72000.000    24225441.784           0.000           0.000    24225442.980    24225441.114   127305563.622           0.000           0.000
95065858.191    97545829.951        1049.393           0.000           0.000        783.702         804.118 -999    -1139611.110    21265266.236    20569607.181     -
111238.265       -1479.328        1434.828       -1565.514 0.000000 2 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0 1 0
1 E09 25734       72000.000    22080411.949           0.000           0.000    22080412.842    22080410.856   116033368.802           0.000           0.000
86648329.798    88908710.053       -1691.396           0.000           0.000       -1263.036       -1295.989 -999   -17845800.349    22900911.983     5827795.911
1806654.687       -383.525         442.044       -2915.585 0.000000 2 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 0 1 0
```

4.5. THE PROCESSLOWLEVEL EXECUTABLE

The `processLowLevel` executable performs the PPP computation using the data read from the `lowlevel` file. It uses the following command:

```
./processLowlevel –conf conf_process.txt –rover rover.txt < lowlevel.txt
```

Where:

- `conf_process.txt` is the general configuration file (the configuration is the same for each rover), refer §4.2.1
- rover.txt contains the name of each receiver (in the same order of the getStream configuration file) and the rover a priori position (“0.0 0.0 0.0” if not used), refer §4.2.2

See §4.2.5 for the description of the lowlevel file. This file can be generated by using the [-lowlevel] option of processStream (refer §4.2) or the generateLowLevel executable.

Here is an example of the output:

```
20-06-16 20:59:01.000 GAMG PPP 17  0  8  8 -3191609.065 +- 0.011  4096901.200 +- 0.015  3691840.481 +- 0.011 2.104149 + 0.050591 +- 0.000701
0.0028
20-06-16 20:59:02.000 GAMG PPP 17  0  8  8 -3191609.065 +- 0.011  4096901.197 +- 0.015  3691840.481 +- 0.011 2.104150 + 0.050589 +- 0.000701
0.0026
20-06-16 20:59:03.000 GAMG PPP 17  0  8  8 -3191609.065 +- 0.011  4096901.198 +- 0.015  3691840.484 +- 0.011 2.104149 + 0.050588 +- 0.000701
0.0025
20-06-16 20:59:04.000 GAMG PPP 17  0  8  8 -3191609.065 +- 0.011  4096901.196 +- 0.015  3691840.485 +- 0.011 2.104149 + 0.050587 +- 0.000701
0.0025
```

From the 1.4 version, the `verbose mode` in the processLowlevel executable program is available in the same way as the one already defined into processStream to harmonize these 2 processes.

5. Description of the algorithms

The different algorithms, models and internal structure of the software are described in the paper joined in this package.

D.Laurichesse, A. Privat, "An Open-source PPP Client Implementation for the CNES PPP-WIZARD Demonstrator", Proceedings of the ION GNSS+ 2015, September 2015, Tampa, Florida

6. Examples

The "tst" directory contains some examples with their configuration files and the associated results. To execute all these examples, use the command: **./test.sh**.

The examples have been created by following these different steps:

- Data acquisition with getStream :
  ```
  getStream < conf_get.txt > brut_gamg.txt
  ```
  To stop getStream, use the kill command with the PID of the getStream process.
  The content of "conf_get.txt" file is :
  ```
  login:password@94.23.202.142:2101/GAMG00KOR0 7 1
  login:password@94.23.202.142:2101/SSRA00CNE0 7 1
  login:password@94.23.202.142:2101/RTCM3EPH-MGEX 7 1
  ```

- PPP computation with processStream:
  Three different PPP computations are performed.
  - Mode PPP_SF with GPS and Glonass:
    ```
    zcat brut_gamg.txt.gz | ../processStream -conf conf_process_PPP_SF_GPSGLO.txt -rover rover_gamg.txt -dcb "*.DCB" >output_PPP_SF_GPSGLO
    ```
  - Mode PPP_AR with GPS and Glonass:
    ```
    zcat brut_gamg.txt.gz | ../processStream -conf conf_process_PPP_AR_GPSGLO.txt -rover rover_gamg.txt -dcb "*.DCB" >output_PPP_AR_GPSGLO
    ```
  - Mode PPP_AR with GPS, Glonass and Galileo:
    ```
    zcat brut_gamg.txt.gz | ../processStream -conf conf_process_PPP_AR_ALL.txt -rover rover_gamg.txt -dcb "*.DCB" >output_PPP_AR_ALL
    ```

- Lowlevel file generation with processStream;
  ```
  zcat brut_gamg.txt.gz | ../processStream -conf conf_process_PPP_AR_ALL.txt -rover rover_gamg.txt -dcb "*.DCB" -lowlevel >output_PPP_AR_GPSGLO
  ```

- PPP computation with processLowlevel:
  ```
  zcat lowlevel_gamg.txt.gz | ../processLowLevel -conf conf_process_PPP_AR_GPSGLO.txt -rover rover_gamg.txt > output_lowlevel 2>/dev/null
  ```

- Lowlevel file generation with generateLowLevel:
  ```
  time ../generateLowLevel -rinex "GAMG*.rnx" -sp3 "cnt21102.sp3" -clk "cnt21102.clk" -atx igs14_2108.atx -dcb "P1*.DCB" -chan channelGlo.txt >meas_gamg.spa
  ```

All input and output files are located in the "tst" directory.