STFProc - Processing of STF based information

Introduction xxx

The SBF format for GNSS receivers from Septentr io contain different `SBF` (Septentrio Binary Format) blocks. Each `SBF` block contains related GNSS information and can be converted to a readable `STF` (Septentrio Text Format) file using the `sbf2stf` program.

The `STF` blocks in a `SBF` file can be obtained by running:

```
$ sbf2stf -f <sbf-file-name>
```

and results in a file similar to:

```
*****************************
Summary of Blocks Information
*****************************
[005914][5914]_Block_Count=     86400
    (ReceiverTime (v1) = Current receiver and UTC time)
[012219][4027]_Block_Count=     86400
    (MeasEpoch (v2) = measurement set of one epoch)
[020384][4000]_Block_Count=     86400
    (MeasExtra (v1) = additional info such as observable variance)
[012199][4007]_Block_Count=     86400
    (PVTGeodetic (v2) = Position, velocity, and time in geodetic coordinates)
[005906][5906]_Block_Count=     86400
    (PosCovGeodetic (v1) = Position covariance matrix (Lat, Lon, Alt))
[005908][5908]_Block_Count=     86400
    (VelCovGeodetic (v1) = Velocity covariance matrix (North, East, Up))
[004001][4001]_Block_Count=     86400
    (DOP (v2) = Dilution of precision)
[004008][4008]_Block_Count=     86400
    (PVTSatCartesian (v1) = Satellite positions)
[012201][4009]_Block_Count=     86400
    (PVTResiduals (v2) = Measurement residuals)
[004011][4011]_Block_Count=     86400
    (RAIMStatistics (v2) = Integrity statistics)
[005935][5935]_Block_Count=     86400
    (GEOCorrections (v1) = Orbit, Clock and pseudoranges SBAS corrections)
[005950][5950]_Block_Count=     86400
    (BaseLine (v1) = Rover-base vector coordinates)
[005921][5921]_Block_Count=     86400
    (EndOfPVT (v1) = PVT epoch marker)
[004013][4013]_Block_Count=     86400
    (ChannelStatus (v1) = Status of the tracking for all receiver channels)
[004012][4012]_Block_Count=     86400
```

```
    (SatVisibility (v1) = Azimuth/elevation of visible satellites)
...
[005894][5894]_Block_Count=       119
    (GPSUtc (v1) = GPS-UTC data from GPS subframe 5)
[004002][4002]_Block_Count=      1610
    (GALNav (v1) = Galileo ephemeris, clock, health and BGD)
[004030][4030]_Block_Count=      4605
    (GALIon (v1) = NeQuick Ionosphere model parameters)
[004031][4031]_Block_Count=      4514
    (GALUtc (v1) = GST-UTC data)
[004032][4032]_Block_Count=      3165
    (GALGstGps (v1) = GST-GPS data)
[004017][4017]_Block_Count=    163936
    (PSRawCA (v1) = GPS CA navigation frame)
[004022][4022]_Block_Count=     67735
    (ALRawFNAV (v1) = Galileo F/NAV navigation page)
[004058][4058]_Block_Count=         4
    (IPStatus (v1) = IP address, gateway and MAC address)
[004019][4019]_Block_Count=     64991
    (PSRawL5 (v1) = GPS L5 navigation frame)
[004023][4023]_Block_Count=    342258
    (ALRawINAV (v1) = Galileo I/NAV navigation page)
[004018][4018]_Block_Count=     50444
    (PSRawL2C (v1) = GPS L2C navigation frame)
[004003][4003]_Block_Count=      4196
    (GALAlm (v1) = Almanac data for a Galileo satellite)
[004034][4034]_Block_Count=      2671
    (ALSARRLM (v1) = Search-and-rescue return link message)
**********************************
Total of 41 Different blocks found
Total of 0 CRC errors found
**********************************
```

For getting one or more of these `STF` blocks, please execute:

```
$ sbf2stf -h
```

This `stfproc` repository currently processes for following `STF` blocks:

- **`stfgeodetic.py`**
  - processing of PVTGeodetic (v2) = Position, velocity, and time in geodetic coordinates
- **`stfrxstatus.py`**
  - processing of ReceiverStatus (v2) = Overall status information of the receiver

Script `stfgeodetic.py`

The script `stfgeodetic.py` reads the PVTGeodetic v2 `STF` file into a `python DataFrame` and

- calculates from the geodetic coordinates the `UTM` projection coordinates
- adds a `DateTime` structure.

The script plots the `UTM` coordinates (versus time and scatter plot), determines what navigation services have been used and whether 2D/3D positioning is used. This is reflected in the plots created.

Getting help

```
$ stfgeodetic.py -h
usage: stfgeodetic.py [-h] [-d DIR] -f FILES -g GNSS [-m MARKER MARKER MARKER]
                      [-l {CRITICAL,ERROR,WARNING,INFO,DEBUG,NOTSET} {CRITICAL,ERROR,WARNIN

stfgeodetic.py reads in a sbf2stf converted SBF Geodetic-v2 file and make UTM
plots

optional arguments:
  -h, --help            show this help message and exit
  -d DIR, --dir DIR     Directory of SBF file (defaults to .)
  -f FILES, --files FILES
                        Filename of PVTGeodetic_v2 file
  -g GNSS, --gnss GNSS  GNSS System Name
  -m MARKER MARKER MARKER, --marker MARKER MARKER MARKER
                        Geodetic coordinates (lat,lon,ellH) of reference point
                        in degrees: ["50.8440152778" "4.3929283333"
                        "151.39179"] for RMA, ["50.93277777", "4.46258333",
                        "123"] for Peutie, default ["0", "0", "0"] means use
                        mean position
  -l {CRITICAL,ERROR,WARNING,INFO,DEBUG,NOTSET} {CRITICAL,ERROR,WARNING,INFO,DEBUG,NOTSET},
                        specify logging level console/file (default INFO
                        DEBUG)
```

Example runs

```
$ stfgeodetic.py -g 'GNSS OS' -d ${HOME}/RxTURP/BEGPIOS/ASTX/19100/stf
    -f SEPT1000.19__PVTGeodetic_2.stf -l INFO DEBUG
$ stfgeodetic.py -g 'Galileo PRS' -d ${HOME}/Nextcloud/E6BEL/19255/stf
    -f STNK2550.19__PVTGeodetic_2.stf -l INFO DEBUG
```

Example of output

A python `DetaFrame` is saved as a `CSV` file, containing the geodetic and UTM position information.

Following plots are created:

Figure 1: UTM coordinates vs time

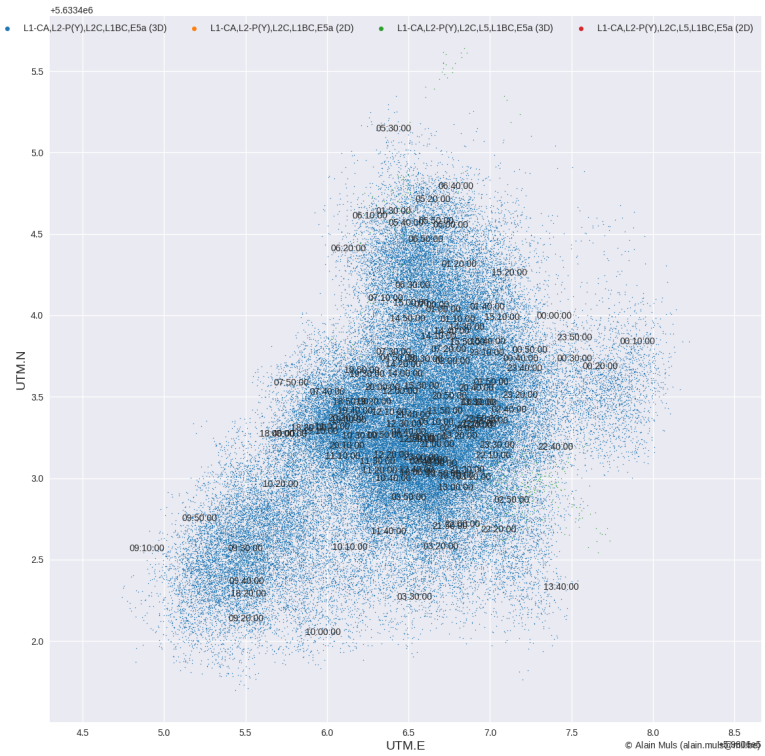

Figure 2: UTM scatter plot

Script `stfrxstatus.py`

The script `stfrxstatus.py` reads the ReceiverStatus v2 `STF` file into a `python DataFrame` and plots the automatic gain control (AGC) of the different front-ends.

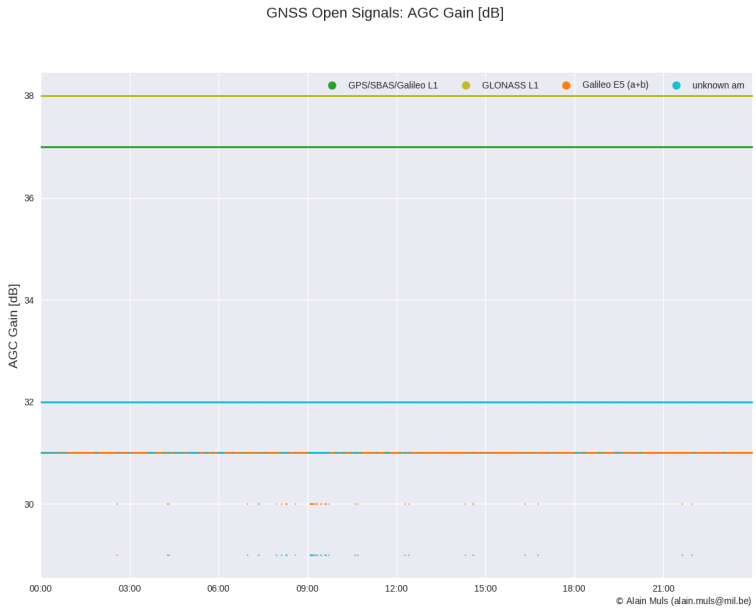

Figure 3: Plot of AGC on front-ends AsteRx SB