

GNSS data handling and the GeoRust rinex crate

Christian Brønnum-Hansen



Styrelsen for Dataforsyning
og Infrastruktur

Greenland



Faroe Islands



Denmark



raw SBF

rinex

raw SBF



Agency for Data Supply and Infrastructure

DTU Space
National Space Institute

research



DATAFORSYNINGEN

public access



EPN

Flow of GNSS data

- different data formats arrive at SDFI from Greenland, Faroe Islands, and Denmark as well as commercial providers
- for Greenland the observation interval relies on the available data connection
- processed at our servers and distributed to
 - DTU
 - dataforsyningen.dk
 - EPN (EUREF Permanent GNSS Network)
- EPN guidelines for reference stations largely sets the standard for outgoing data



Agency for Data Supply
and Infrastructure

GORM

GNSS Operations, Register and Monitoring system

- old (read: thoroughly tested)
- developed in PERL
- relies on closed-source software

Edit site AAS200GRL

Parameter	Value
Site	AAS200GRL
Site 4ch	AAS2
Markernumber	43010M002
Markertype	GEODETIC
Freq	Hourly
Datasource	Receiver
Interval	1
Observer	DTU Space
Agency	SDFI
Siteflags	0
Active	<input checked="" type="checkbox"/>

[Save](#) [Back to sitelist](#) [Edit destinations](#) [Edit antennas](#) [Edit receivers](#) [Edit positions](#)

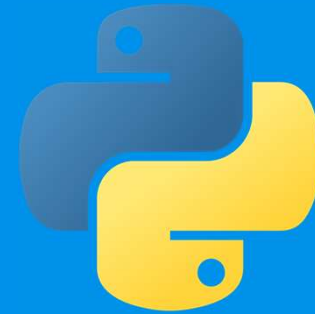
Site 4ch must match first 4 letters on incoming files.
If Marker Number is blank and Marker Number is Unknown in original file, set Marker Number to short sitename.
If Marker number is blank and Marker number is set in original file, do not change original.
if Marker number is, always redefine Marker number in file.
Position is only altered if specified.
Observer and Agency defaults to *SDFI*.
Siteflags bits. Bit#1=Priority-site. Restart of jobengine required.

[Goto Main Menu](#)

SITE	89	90	91	92	93	94	95	96	97	98
AAS200GRL	97	97	97	97	96	97	97	97	97	97
ASRY00GRL	98	98	98	98	95	98	98	98	97	98
BLAS00GRL	99	99	98	99	96	99	99	99	98	99
DANR00GRL	96	96	96	96	95	96	96	96	95	96
DOS000GRL	92	92	92	92	99	92	92	92	92	92
DRK000GRL	95	95	95	95	92	95	95	95	92	95
QPR000GRL	99	99	99	99	96	99	99	99	98	99
QRC000GRL	97	97	97	97	94	97	97	97	96	97
REL200GRL	93	93	93	93	92	93	93	93	93	93
RJKB00GRL	97	97	97	97	94	97	97	97	96	97
RHS000GRL	91	90	92	92	89	92	92	92	91	91
RHD000GRL	97	97	97	97	94	97	97	97	96	97
ILUL00GRL	95	95	95	95	94	94	94	95	95	95
ISGR00GRL	97	97	97	97	95	96	96	97	96	97
JOB100GRL	99	99	99	99	96	99	99	99	97	99
JNL100GRL	94	95	95	95	92	94	95	95	94	95
KAG000GRL	95	96	96	96	93	96	96	96	96	95
KAG200GRL	97	97	97	97	94	97	97	97	96	97
KMP100GRL	90	90	90	89	89	89	89	90	90	90
KRS000GRL	92	92	92	92	89	92	92	92	92	92
KEL100GRL	92	92	92	92	89	92	92	92	91	92
KLQ300GRL	98	99	98	98	97	98	98	98	98	98
KLQ000GRL	98	98	98	98	97	98	98	98	98	98
KLY200GRL	98	97	98	98	95	98	98	98	97	97
KNS000GRL	96	96	96	96	93	96	96	96	96	96
KPK000GRL	97	97	97	97	94	97	97	97	96	97
KSB000GRL	94	94	94	94	91	94	94	94	93	94
KSF000GRL	96	96	96	96	94	96	96	96	95	96
KUNQ00GRL	95	95	95	95	92	95	95	95	95	95
KUL000GRL	96	96	96	96	94	96	96	96	96	96
KUL100GRL	91	92	92	91	91	91	91	92	91	92
LEB000GRL	98	98	99	99	96	99	99	99	97	99
LEF000GRL	97	97	97	97	94	97	97	97	96	97
LYB000GRL	97	97	97	97	94	97	97	97	97	97
MAB000GRL	98	98	98	98	95	98	98	98	97	98
MIR200GRL	96	96	96	96	93	96	96	96	95	96
MSV000GRL	95	95	95	95	92	95	95	95	94	95
MPY000GRL	98	98	98	98	97	98	98	98	97	98
MKD000GRL	100	99	99	99	96	99	99	100	99	100
MRE000GRL	98	98	98	98	95	98	98	98	97	98
MFK000GRL	98	98	98	98	96	97	97	97	96	97
PANI00GRL	98	98	98	98	96	97	97	98	98	98
PLPK00GRL	85	85	85	86	85	86	85	85	85	85
PAR000GRL	94	94	94	94	91	94	94	94	94	94

Goals for data handling: scalability and maintainability

- Python for maintainability
 - concurrency (asyncio)
- containerized for easy deployment (Docker)
- central database (postgresql)
- open-source
 - version control with git
 - external dependencies



Software dependencies for data processing

Processing step	Current software	Open?	OSS alternatives (non-exhaustive list)
SBF to rinex (ver. 3)	sbf2rin [1]	✘	? (some projects)
<i>decimate</i>	gfzrn [2]	✘	BNC [4], GeoRust rinex [5]
gap analysis	GORM		GeoRust rinex
<i>file splicing</i>	gfzrn	✘	BNC, GeoRust rinex
<i>quality check</i>	g-nut/anubis [3]	✘	BNC, GeoRust rinex

[1] <https://www.septentrio.com/en/products/gps-gnss-receiver-software/rxtools>

[2] <https://gnss.gfz-potsdam.de/services/gfzrn>

[3] <https://gnutsoftware.com/software/anubis/>

[4] <https://igs.bkg.bund.de/ntrip/bnc>

[5] <https://github.com/georust/rinex>

W

• T

• r

• M

• H

```

e340115@gpsftp7: ~/gitrepos/my_rust_examples
#[derive(Clone, Debug)]
struct Coord(f32, f32);

fn add_by_owing(p1: Coord, p2: Coord) -> Coord {
    Coord(p1.0 + p2.0, p1.1 + p2.1)
}

fn main() {
    let p1 = Coord(3.142, 2.718);
    let p2 = Coord(1.414, 1.618);
    let p3 = add_by_owing(p1.clone(), p2.clone());
    println!("{p1:?} + {p2:?} = {p3:?}");
}

"add_coords_owing.rs" 13L, 313B written                                1,15      A11
e340115@gpsftp7:~/gitrepos/my_rust_examples$ rustc add_coords_owing.rs
e340115@gpsftp7:~/gitrepos/my_rust_examples$ ./add_coords_owing
Coord(3.142, 2.718) + Coord(1.414, 1.618) = Coord(4.5559998, 4.336)
e340115@gpsftp7:~/gitrepos/my_rust_examples$

e340115@gpsftp7: ~/gitrepos/my_rust_examples
#[derive(Debug)]
struct Coord(f32, f32);

fn add_by_borrowing(p1: &Coord, p2: &Coord) -> Coord {
    Coord(p1.0 + p2.0, p1.1 + p2.1)
}

fn main() {
    let p1 = Coord(3.142, 2.718);
    let p2 = Coord(1.414, 1.618);
    let p3 = add_by_borrowing(&p1, &p2);
    println!("{p1:?} + {p2:?} = {p3:?}");
}

"add_coords_borrowing.rs" 13L, 301B                                1,1      A11
e340115@gpsftp7:~/gitrepos/my_rust_examples$ rustc add_coords_borrowing.rs
e340115@gpsftp7:~/gitrepos/my_rust_examples$ ./add_coords_borrowing
Coord(3.142, 2.718) + Coord(1.414, 1.618) = Coord(4.5559998, 4.336)
e340115@gpsftp7:~/gitrepos/my_rust_examples$

[0] 0: bash"                                "gpsftp7" 14:41 08-Mar-24

```

[1] <https://survey.stackoverflow.co/2023/#section/admired-and-desired-programming-scripting-and-markup-languages>

[2] <https://google.github.io/comprehensive-rust/cargo/rust-ecosystem.html>

[3] <https://benchmarksgame-team.pages.debian.net/benchmarksgame/fastest/rust-gpp.html>

The GeoRust ecosystem



Handling GIS data formats

GDAL	Bindings for the Geographic Data Abstraction Library (GDAL) for reading and writing raster and vector GIS files.	GitHub	crates.io
GeoJSON	Work with GeoJSON files.	GitHub	crates.io
GPX	Work with GPS files.	GitHub	crates.io
GeoTIFF	Work with GeoTIFF raster files.	GitHub	crates.io
KML	Work with KML files.	GitHub	crates.io
netCDF	Bindings for Network Common Data Form (netCDF) library. Can read and write HDF5 files.	GitHub	crates.io
OSM	Work with the OpenStreetMaps PBF files.	GitHub	crates.io
Shapefile	Work with shape (SHP) files.	GitHub	crates.io

TileJSON	Work with TileJSON files.	GitHub	crates.io
Transit	Work with GTFS files.	GitHub	crates.io
WKT	Work with Well-Known Text (WKT) files.	GitHub	crates.io
World-file	Work with World-files.	GitHub	crates.io
STAC	Work with SpatioTemporal Asset Catalogs (STAC)	GitHub	crates.io
PgSTAC	Read from and write to <code>pgstac</code> databases.	GitHub	crates.io
OGC API	OGC API building blocks	GitHub	crates.io
RINEX	Read, parse and generate RINEX files. Can read RINEX4 files.	GitHub	crates.io
SP3	Read, parse and generate SP3 files (high precision orbits).	GitHub	crates.io
e.		GitHub	crates.io

[1] <https://georust.org>

GeoRust rinex_[1] operations: decimation and splicing

- decimation with gfzrnrx ~1s

```
e340115@gpsftp7:~/gitrepos/georust_rinex_examples$ time gfzrnrx -f inp files/SKG100DNK_R_20240510000_01H_01S_MO.rnx -fout ::RX3:: -  
smp 30 -f -q -kv  
real    0m0.866s  
user    0m0.801s  
sys     0m0.064s
```

- with BNC ~15s

```
e340115@gpsftp7:~/gitrepos/georust_rinex_examples$ time bnc --nw --conf /dev/null --key reqcAction Edit/Concatenate --key reqcObsFile files/SKG100DNK_R_20240510000_01H_01S_MO.rnx --key reqcOutObsFile bnc_decimate.rnx --key reqcRnxVersion 3 --key reqcSampling 30  
real    0m15.852s  
user    0m13.690s  
sys     0m0.077s
```

- with GeoRust switch to a stream-based approach desired

```
e340115@gpsftp7:~/gitrepos/georust_rinex_examples$ time ./a.out  
Epochs 3600  
decimated to 120  
real    0m3.382s  
user    0m2.906s  
sys     0m0.445s
```

- GeoRust uses less memory than gfzrnrx

```
e340115@gpsftp7:~/gitrepos/georust_rinex_examples$ time ./a.out  
Epochs 3600  
decimated to 120  
real    0m0.064s  
user    0m0.047s  
sys     0m0.016s
```

- splicing with gfzrnrx ~23s

```
e340115@gpsftp7:~/gitrepos/georust_rinex_examples$ time gfzrnrx -f inp files/SKG100DNK_R_20240510000_01H_01S_MO.rnx files/SKG100DNK_R_20240510100_01H_01S_MO.rnx -fout splice.rnx -f -q -kv -splice_direct  
real    0m23.446s  
user    0m23.101s  
sys     0m0.336s
```

- with BNC ~45s

```
e340115@gpsftp7:~/gitrepos/georust_rinex_examples$ time bnc --nw --conf /dev/null --key reqcAction Edit/Concatenate --key reqcObsFile files/SKG100DNK_R_20240510000_01H_01S_MO.rnx files/SKG100DNK_R_20240510000_01H_01S_MO.rnx --key reqcOutObsFile bnc_splice.rnx  
real    0m45.447s  
user    0m43.004s  
sys     0m0.412s
```

- GeoRust uses less memory than gfzrnrx when merging 24 one-hour files

[1] GeoRust RINEX Team (2023), RINEX: analysis and processing (Apache-2/MIT), <https://georust.org>

GeoRust rinex operations: quality analysis

- aims for teqc-like quality check with support for new rinex format

```

first epoch  last epoch  hrs  dt  #expt  #have  %  mp1
SUM 10  4 30 00:00 10  4 30 23:59 24.00  30  41538  41293  99  0.39
    
```

- excerpts from GeoRust

RINEX Quality Check summary

Version	rinex-qc: v0.1.10
Context	
File	Name
Observations	ARGI00FRO_R_20240600000_01D_30S_MC
Broadcast Navigation	ARGI00FRO_R_20240600000_01D_MN.rnx
ANTEX	None
SP3	None

Epochs	
Total#	2880
w/ observations	2880 (100%)
Complete	Epochs with at least Phase + PR in dual frequency, with both SNR and elev above masks
	G12 L2/L1 1009 (35%)
	G05 L2/L1 1051 (36%)
	G22 L2/L1 979 (33%)
	G06 L2/L1 1075 (37%)
	G13 L2/L1 790 (27%)
	G03 L2/L1 1029 (35%)
	G09 L5/L1 1024 (35%)
	G19 L2/L1 1040 (36%)
	G28 L5/L1 1063 (36%)
	G31 L2/L1 1043 (36%)
	G29 L2/L1 1045 (36%)
	G07 L2/L1 1091 (37%)
	G26 L5/L1 1040 (36%)
	G25 L2/L1 992 (34%)

analysis

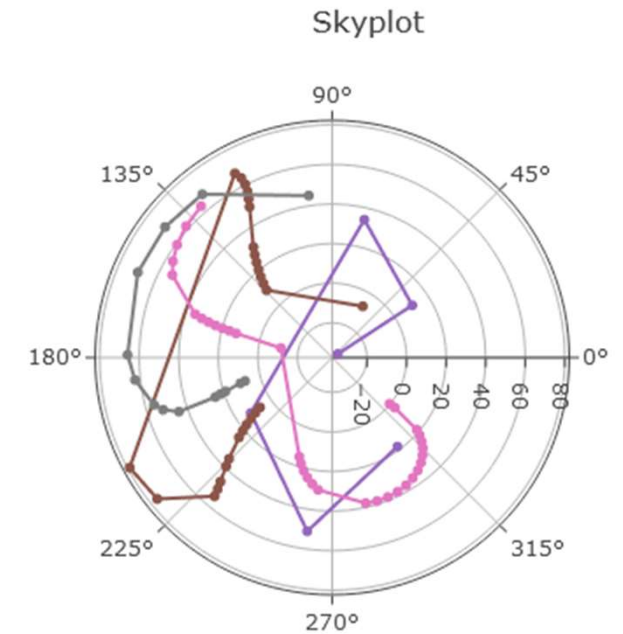
ng

2-28T23:59:23 U

rate (Header)

ant Sample rate 30 s (0.033 Hz)

analysis **No gaps detected**



[1] <https://www.unavco.org/software/data-processing/teqc/teqc.html>

GeoRust rinex is (promising) work in progress

- promising with plenty functionality, but more work needed
- on-going development
- get involved?
 - coding
 - testing
 - feature requests

