SRON
Netherlands Institute for Space Research

SPEXone

SPEXone Level 1A to 1C Processor Release Notes

Raul Laasner, Jeroen Rietjens, Otto Hasekamp, Martijn Smit, Guangliang Fu, Richard van Hees, Jochen Landgraf

Citation: Laasner, R., Aan de Brugh J.M.J., Rietjens J., Hasekamp O.P., Smit, M., Van Hees, R.M., Landgraf, J.: SPEXone Level 1A to 1C Processor Release Notes, SRON Netherlands Institute For Space Research, Utrecht, The Netherlands, SRON-SPEXoneL1-2023-02, issue: 1.0, 2023

document number	:	SRON-SPEXoneL1-2023-10-06
software version	:	2.4
date	:	2023-07-14
status	:	final

Contents

1 Introduction

The SPEXone data processor consists of three parts: the L1A-L1B processor, the L1B-L1C processor, and a calibration key data (CKD) generator. The CKD generator uses dedicated on-ground calibration measurements to acquire the CKD. The L1A-L1B processor uses flight L1A data together with the CKD to generate L1B data. The L1B-L1C data collocates L1B data on a common geolocation grid for a predefined reference height.
The L1A-L1B processor and the CKD generator are algorithmically interlinked to ensure full consistency of the derived CKD and the calibration of the L1A data. Therefore, it has been decided to include both elements in the same software package even though the CKD generator is not required for processing flight data.

2 New features

2.1 L1B parallelism

The L1A-L1B module of the processor now makes use of MPI remote memory access (RMA), i.e. shared memory. This incurs minimal communication between MPI ranks and is expected to run faster.

3 Installation and running

3.1 Dependencies

- GNU C++ compiler version 11
- CMake version 3.18
- NetCDF C and C++ libraries
- HDF libraries
- Linear algebra library (e.g. lapack but preferably Intel MKL)
- Intel FFT library (optional) — if not available a fallback library is used

3.2 Data

This delivery includes two test cases:

- A simulated orbit that is partially populated with data from high-accuracy simulations. Realistic flight data spans the extent of 3 L1C granules (about 15 minutes). The rest of the orbit is a measured dark signal as seen in Fig. 1.
- A simulated orbit fully populated with data. However, these comes from less accurate simulations. Furthermore, the data is undefined close to the equator as seen on the right side of Fig. 1 at timestamps 4000–5000.

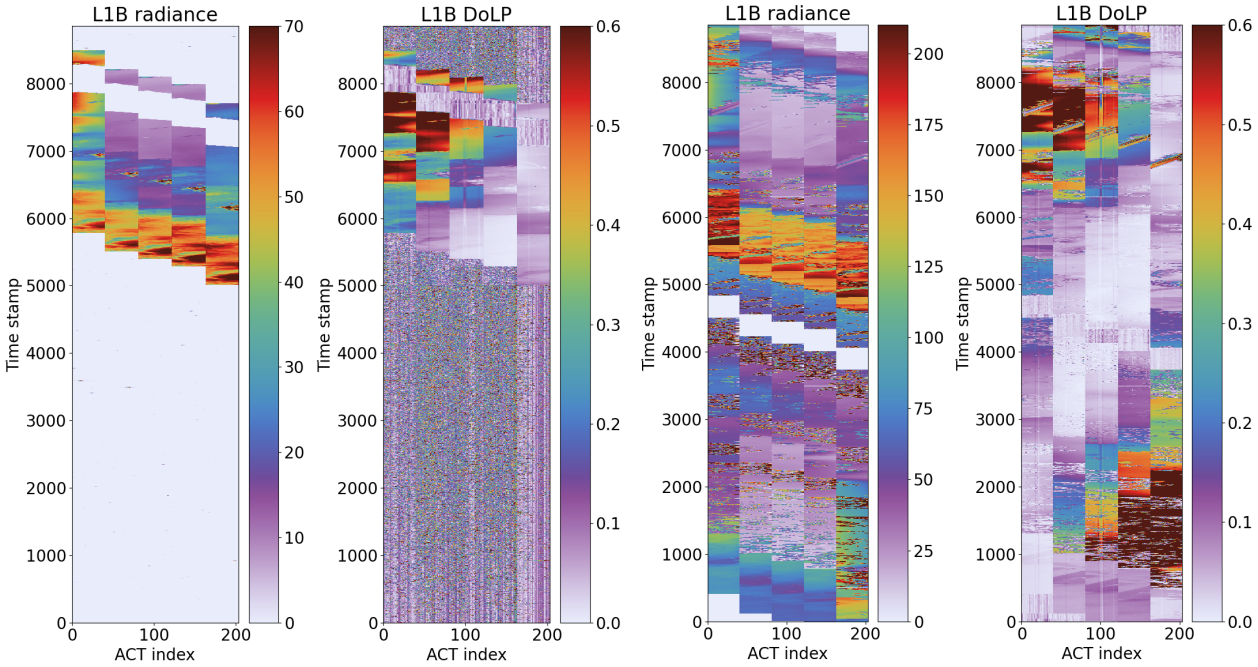

Figure 1: Radiance and DoLP from L1B products as a function of ACT and timestamp indices. Data is shown for all five viewports for a fixed wavelength of 556.1 nm. Left: radiance and DoLP of the accurate “3 granule” orbit. Right: radiance and DoLP of the full “10 granule” orbit.

In the file names, whether it’s data products or configuration files, “3 granules” refers to the orbit containing high-accuracy simulation data covering 3 L1C granules, and “10 granules” refers to the other orbit built from less accurate simulations covering the whole orbit.

3.3 Building

Make a copy of the initial CMake cache file

```
cp initial_cache.cmake.example initial_cache.cmake
```

found in the root source directory and edit it to reflect your environment. Next, create a build directory for configuring and building the SPEXone executable. From the root directory the procedure could look like this:

```
mkdir build && cd build
cmake -C ../initial_cache.cmake -D CMAKE_BUILD_TYPE=release ..
make -j
```

3.4 Running

Start by calibrating both L1A products:

```
mpirun -np 40 <spexone> l1b_3_granules.yaml
mpirun -np 40 <spexone> l1b_10_granules.yaml
```

where `<spexone>` is the SPEXone processor executable. The generated L1B products may be compared with the reference ones:

```
h5diff l1b_3_granules.nc l1b_3_granules_ref.nc
h5diff l1b_10_granules.nc l1b_10_granules_ref.nc
```

Next, generate L1C products using three L1C grid files:

```
mpirun -np 8 <spexone> l1c_20220321T115708_3_granules.yaml
mpirun -np 8 <spexone> l1c_20220321T120208_3_granules.yaml
mpirun -np 8 <spexone> l1c_20220321T120708_3_granules.yaml
mpirun -np 8 <spexone> l1c_20220321T115708_10_granules.yaml
mpirun -np 8 <spexone> l1c_20220321T120208_10_granules.yaml
mpirun -np 8 <spexone> l1c_20220321T120708_10_granules.yaml
```

Compare the results with reference files:

```
h5diff l1c_20220321T115708_3_granules.nc l1c_20220321T115708_3_granules_ref.nc
h5diff l1c_20220321T120208_3_granules.nc l1c_20220321T120208_3_granules_ref.nc
h5diff l1c_20220321T120708_3_granules.nc l1c_20220321T120708_3_granules_ref.nc
h5diff l1c_20220321T115708_10_granules.nc l1c_20220321T115708_10_granules_ref.nc
h5diff l1c_20220321T120208_10_granules.nc l1c_20220321T120208_10_granules_ref.nc
h5diff l1c_20220321T120708_10_granules.nc l1c_20220321T120708_10_granules_ref.nc
```

4 Files

The delivery is accessible at `https://public.spider.surfsara.nl/project/spexone/PACE/L1A-L1C/2023_07_14/`.

- `spexone_cal.tar.gz` — source code. Unpack and compile according to instructions in Sec. 3.3.
- `l1a_3_granules.nc` — L1A product (high-accuracy)
- `l1a_10_granules.nc` — L1A product
- `l1b_3_granules.yaml` — Configuration file for generating an L1B product from `l1a_3_granules.nc`
- `l1b_10_granules.yaml` — Configuration file for generating an L1B product from `l1a_10_granules.nc`
- `l1b_3_granules_ref.nc` — Reference L1B product. Should be identical to that produced by running `l1b_3_granules.yaml`
- `l1b_10_granules_ref.nc` — Reference L1B product. Should be identical to that produced by running `l1b_10_granules.yaml`
- `l1c_*_3_granules.yaml` — Configuration files for generating L1C products using either `l1b_3_granules.nc` or `l1b_3_granules_ref.nc` and the L1C grid files as input
- `l1c_*_10_granules.yaml` — Configuration files for generating L1C products using either `l1b_10_granules.nc` or `l1b_10_granules_ref.nc` and the L1C grid files as input
- `l1c_*_3_granules_ref.nc` — Reference L1C files. Should be identical to those produced by the `l1c_*_3_granules.yaml` configuration files.
- `l1c_*_10_granules_ref.nc` — Reference L1C files. Should be identical to those produced by the `l1c_*_10_granules.yaml` configuration files.
- `PACE_SPEX.*.L1C.nc` — L1C grid files used for the L1B-L1C runs.
- `ckd.nc` — calibration key data required for the L1A-L1B runs
- `gebco_ocssw_v2020.nc` — elevation map required for geolocation
- `SPX1_CKD_BIN_TBL_20210304T124000_001.nc` — binning table
- `binned_fs_hybrid_reference_spectrum_c2022-11-30_with_unc.nc` — solar spectrum. This will be stored in L1B and L1C products after convolving with the ISRF.