The Journal of Open Source Software

Bitstream – Binary Data for Humans

Sébastien Boisgérault[1]

1 MINES ParisTech, PSL Research University, Centre for robotics

DOI: 10.21105/joss.00541

Software

- Review
- Repository
- Archive

Submitted: 16 November 2017
Published: 31 January 2018

Summary

Audiophiles are familiar with multiple digital audio file formats (WAV, MP3, AAC, ALAC, FLAC, etc.) and generally know that this multiplicity is justified by different trade-offs and features (in terms of quality, compression rate, complexity, for example). The same logic drives the research for new binary formats in many contexts. Such research goes through an experimental phase where the development of codecs – the software that transforms back and forth the original data into binary data – is required for any theoretical design. Any tool which can simplify and speed up the prototyping of such codecs therefore improves significantly this iterative process.

In this context, Bitstream (Boisgérault 2017a) provides a Python library with a simple, high-level and customizable programming interface to manage binary data. Many classic but menial tasks usually required are automatically taken care of under the hood.

The cornerstone of the library is the use of the "bitstream" abstraction. The "stream" part means that we use a simple model where one can only write data at one end of the binary structure and read data at the other end, in the same order. The "bit" part means that the library can work seamlessly with individual bits and not merely bytes, a feature frequently required by lossless data compression schemes. Bitstream supports out of the box data types from Python and NumPy: ASCII strings, (arrays of) booleans, fixed-size integers, floating-point numbers, etc. One can also define and register custom (even parametrized) types and their binary representation, and then use them with the same interface. Since the library supports creation and restoration of stream snapshots, it's possible to go beyond the stream model when necessary; this "time machine" scheme is more than adequate for many use cases (header lookahead, decoders with strong exception safety, etc.). And since bitstream is a Python C extension, it is fast enough for many applications.

The design of bitstream was initially driven by the development of a "Digital Audio Coding" graduate course at MINES ParisTech University (MINES ParisTech, PSL University 2013; Boisgérault 2017b). In this context, which mixes information theory, binary formats and numeric data, the bitstream abstraction works really well. A simple interface was required to replace pseudo-code with actual code, bridging the gap between lectures and lab sessions. Since none of the Python libraries we were aware of (Python 2.7 Standard Library 2017b, 2017a; Griffiths 2014; Schnell 2013, etc.) supported the feature set described above, bitstream was born.

The library later proved to be useful to prototype and document quickly classic and experimental binary formats and codecs. It was integrated as a component of the Python `audio` package and used by audio coding applications, such as `audio.wave`, a reader and writer of WAVE files (see e.g. Kabal 2017) integrated with NumPy and `audio.shrink`, an experimental lossless codec inspired by SHORTEN (Robinson 1994).

References

Boisgérault, Sébastien. 2017a. "Bitstream: Binary Data for Humans." *Github*. https://github.com/boisgera/bitstream.

———. 2017b. "Digital Audio Coding." *Eul.Ink*. http://eul.ink/audio/.

Griffiths, Scott. 2014. "Bitstring: A Python Module to Help You Manage Your Bits." *Github*. http://scott-griffiths.github.io/bitstring/.

Kabal, Peter. 2017. "Audio File Format Specification." *MMSP Lab, ECE, McGill University*.

MINES ParisTech, PSL University. 2013. "Digital Audio Coding." *MINES ParisTech Graduate School Course Catalog*. https://sgs.mines-paristech.fr/prod/sgs/ensmp/catalog/course/detail.php?code=S1916&year=3a&lang=EN.

Python 2.7 Standard Library. 2017a. "Array — Efficient Arrays of Numeric Values." https://docs.python.org/2/library/array.html.

———. 2017b. "Struct — Interpret Strings as Packed Binary Data." https://docs.python.org/2/library/struct.html.

Robinson, Tony. 1994. "SHORTEN: Simple Lossless and Near-Lossless Waveform Compression." http://mi.eng.cam.ac.uk/reports/svr-ftp/auto-pdf/robinson_tr156.pdf.

Schnell, Ilan. 2013. "Bitarray: Efficient Array of Booleans for Python." *Github*. https://github.com/ilanschnell/bitarray.