**Embedded data format**

формат сериализации двоичных данных для контроллеров или встраиваемых систем.

* Простая и компактная реализация, чтобы было легко портировать на любой язык
* Функционирование без динамического выделения, в малом объёме статически выделенной оперативной памяти 256-1024 байт
* Последовательная запись, без случайного доступа на запись. Требуется для записи напрямую в флэш-память без необходимости стирать страницы памяти или обновлять файл встраиваемой ФС (перезапись в середине тоже файла очень накладная).
* Встроенная Схема данных в виде Си-подобные структур (заголовок данных / таблицы бд)
* Возможность записи нескольких экземпляров данных (строки в таблице бд)
* Поддержка юникода
* Бинарное (блочное) представление и текстовое, для удобства просмотра, +конвертеры между ними
* Бинарная запись данных должна быть как можно более компактной и быстрой для использования при ограниченном количестве памяти и процессора
* little-endian

**ещё не делал**

* парсер (конвертер) из текстового представления в бинарный
* Контроль целостности (блочный, опционально) – CRC16 в конце каждого бинарного блока
* Сжатие целых чисел – ещё не доделал, реализовано только в версии C#

[Variable-length quantity - Wikipedia](https://en.wikipedia.org/wiki/Variable-length_quantity) , [Encoding | Protocol Buffers Documentation](https://protobuf.dev/programming-guides/encoding/#varints) , [Compact varint — уникальность и большие значения за ту же стоимость / Хабр](https://habr.com/ru/articles/350796/)

* Разностное сжатие – сжатие данных, которые мало отличаются между измерениями + Variable-length, например время UNIX (int64 в секундах c 1970), время 1757582277(8 байт), 1757582278(8 байт) – 16 байт заменять на: 1757582277(8 байт), +1(1 байт) – 9 байт

**Недостатки**

* нужно или грузить в память целиком, или обрабатывать в несколько проходов, если он не влезает в память. Также, если нужна не вся информация из файла — все равно придется прочитать его весь или создавать индексы (ссылки на данные) отдельным файлом

**использование конвертера edf.exe**

edf.exe 1D.D t – конвертер динамограммы в **текстовый** вариант 1D.**t**df

edf.exe 1E.E t – конвертер эхограммы в **текстовый** вариант 1E.**t**df

edf.exe 1DAT.DAT t – конвертер данных манометра (в т.ч. погружного) в **текстовый** вариант 1E.**t**df

edf.exe 1D.D b – конвертер динамограммы в **бинарный** вариант 1D.**b**df

edf.exe 1E.E b – конвертер эхограммы в **бинарный** вариант 1E.**b**df

edf.exe 1DAT.DAT b – конвертер данных манометра (в т.ч. погружного) в **бинарный** вариант 1E.**b**df

edf.exe 1D.bdf d – конвертер бинарного в данные эхограммы 1D.**D**

edf.exe 1E.bdf e – конвертер бинарного в данные эхограммы 1E.**E**

edf.exe 1DAT.bdf dat – конвертер бинарного в данные манометра 1DAT.**DAT**

**примеры**

dynChart.xlsx, echoChart.xlsx – построенные динамограмма и эхограмма из примеров файлов

npp\_userDefineLang.xml – раскраска текстового представления для notepad++

t\_write.tdf – пример структур

**ссылки по теме**

[Comparison of data-serialization formats - Wikipedia](https://en.wikipedia.org/wiki/Comparison_of_data-serialization_formats" \l "Comparison_of_binary_formats)

[Log ASCII standard - Wikipedia](https://en.wikipedia.org/wiki/Log_ASCII_standard?ysclid=maul3tojcv75384715)

[CDF Home Page](https://cdf.gsfc.nasa.gov/)

[protocolbuffers/protobuf: Protocol Buffers - Google's data interchange format](https://github.com/protocolbuffers/protobuf/tree/main)

[FlatBuffers Docs](https://flatbuffers.dev/)

[fraillt/bitsery: Your binary serialization library](https://github.com/fraillt/bitsery)

[liteserver/binn: Binary Serialization](https://github.com/liteserver/binn)

[CBOR - Wikipedia](https://en.wikipedia.org/wiki/CBOR) [CBOR — новый бинарный формат представления данных / Хабр](https://habr.com/ru/articles/208690/)

[RIFF — Википедия](https://ru.wikipedia.org/wiki/RIFF)

[EBML — Википедия](https://ru.wikipedia.org/wiki/EBML)

[Efficient XML Interchange (EXI) Format 1.0 (Second Edition)](https://www.w3.org/TR/exi/)

мы далеко не первые

[Полноценная бинарная замена XML / Хабр](https://habr.com/ru/articles/270469/)

[Tree — убийца JSON, XML, YAML и иже с ними / Хабр](https://habr.com/ru/articles/248147/)

