

MoonProtobuf

Protocol Buffers Runtime Encoder / Decoder + Proto Parser for MoonBit

2026 MoonBit 国产开源生态竞赛（个人赛） | MIT License

// 01. 基本信息

项目	MoonProtobuf — Protocol Buffers 独立运行时编解码码库
仓库	GitHub: github.com/Duan-lang-dev/MoonProtobuf GitLink: gitlink.org.cn/Duan525/moonproto
方向	MoonBit 基础库 / 序列化协议 原创实现（基于 Protobuf 公开规范） MIT

// 02. 项目简介

MoonProtobuf 是一个纯 MoonBit 实现的 Protocol Buffers 运行时编解码码库，提供完整的 wire format 支持，特色是附带 .proto 文件解析器。与官方 moonbitlang/protobuf（代码生成器配套运行时）形成互补：官方库面向需要 protoc 工具链的标准 protobuf 工作流，本项目面向轻量序列化、proto 文件工具开发和 protobuf 编码学习。零外部工具依赖、MIT 许可、分层 API 设计，即插即用。

// 03. 核心模块

- >> 底层 — Varint / Wire（166 行）
 - ULEB128 varint 编解码（ $0 \sim 2^{64}-1$ ），含 decode_varint/decode_sint32/decode_sint64 互补 API
 - Zigzag 32/64 有符号映射，小绝对值负数仅占 1 字节
 - 4 种 wire type 读写（Varint 0 / Fixed64 1 / LEN 2 / Fixed32 5），均带负 pos 边界检查
- >> 中层 — Encoder / Decoder（420 行）
 - Encoder: 20 个方法覆盖全部标量类型 + string/bytes/message，Builder 模式链式调用
 - Decoder: 19 个方法，read_tag 驱动循环、按 field_number 分发、skip_field 跳过未知、read_message 嵌套解码
- >> 上层 — Packed Repeated / Proto Parser（555 行）
 - Packed: 5 编码 + 3 解码函数，支持 varint/fixed32/fixed64/float/double 及空数组
 - Proto Parser: proto3 完整语法，message/enum/嵌套类型/字段标签，全部标量类型 + 自定义类型引用 — 官方库无此功能

// 04. 与官方库的差异化定位

维度	moonbitlang/protobuf	MoonProtobuf
定位	protoc 代码生成器配套	独立运行时 + proto 解析器
工作流	protoc-gen-mbt 代码生成	纯 MoonBit 代码调用
API	Writer/Reader trait（流式）	Encoder/Decoder（Builder）
Proto 解析	无	完整 proto3 解析器
异步 I/O	AsyncWriter/Reader	同步 Buffer/Bytes
许可证	Apache-2.0	MIT
外部依赖	protoc + protoc-gen-mbt	零外部工具依赖

互补关系：官方库解决"在 MoonBit 中标准使用 protobuf"的问题，本项目解决"不依赖工具链直接操作 wire format"和"解析 proto 文件"的问题。两者面向不同场景，不构成重复。

// 05. 项目规模

模块	源码	测试	数
varint	87	111	16
wire	79	76	11
encoder	132	83	11
decoder	288	156	14
packed	95	109	6
proto_parser	460	121	8
入口 / 集成测试	13	406	8
CLI	62	—	—
合计	1,216	1,062	80

总计 2,279 行 MoonBit 代码，80 个测试全部通过，22 次有效提交。CI 已配置（check + test + build），GitHub + GitLink 双仓库已推送，MIT 许可证。

// 06. 适用场景

- 轻量序列化 — 无需 protoc 工具链，项目内直接调用 Encoder/Decoder
- Proto 文件工具 — 利用内置解析器读取 .proto 文件做静态分析、文档生成
- 学习与教育 — 分层 API 展示 protobuf wire format 每一层编码细节
- 嵌入式 / IoT — 同步 API + 零工具链依赖，适合资源受限场景
- 跨语言交换 — 与 Go/Rust/Java/Python 等语言的 protobuf 实现兼容