

Starting with Protobufs

What are protobufs?

[Protocol Buffers](#) (protobufs) are a way to define structured messages in `.proto` files.

The point is simple: define the message format once, generate code for your language, and now everyone agrees what the bytes mean (without inventing a new packet format every semester).

Where they live

The source of truth for RoboTeam message definitions is: [RoboTeamTwente/ERC-Protobufs](#).

Other repositories (embedded, software, tooling) should **consume** these definitions (often via a git submodule) instead of copy-pasting `.proto` files around.

Editing rules (the important part)

Field numbers are the real API

In protobuf, the **field number** is what goes on the wire. Renaming a field is usually harmless; changing its number is usually the problem.

Never renumber an existing field unless you are intentionally breaking compatibility and coordinating the update across all consumers.

Safe changes (usually)

- Add a new field with a **new** unused number.
- Add new enum values (don't reuse old numeric values for new meanings).
- Stop using a field before deleting it, and reserve its number if your style guide does that.

Changes that need extra care

- Changing a field type (e.g. `int32` → `string`).
- Changing semantics/units (mm vs m, degrees vs radians).

Typical workflow

1. Edit / add `.proto` files in `ERC-Protobufs`.
2. Generate / update code as required by your project (this step is repo-specific).
3. Update consumers to use the new fields (and handle missing fields safely).
4. Test at least one real send/receive path.
5. Open a PR that lists: what changed, field numbers, and compatibility expectations.

Troubleshooting

“My message doesn’t parse”

- Confirm both sides use the same (or compatible) `.proto` definitions.
- Check field numbers: no reuse, no renumbering.
- Check transport framing (length prefix / delimiter / CRC). Protobuf gives bytes; transport still matters.

“It parses, but values are nonsense”

- Check units and semantics (the most common “it’s wrong but not broken” bug).
- Make sure generated code is up to date (stale generated files cause creative failures).

Protobufs are boring on purpose. Boring schemas prevent exciting debugging sessions. Keep changes small, reviewable, and compatible.

Revision #4

Created 2026-04-15 14:17:17 UTC by Shishir Nambiar

Updated 2026-04-16 13:15:47 UTC by Dmytro Khorsun