

Arm Board Protobuffers

This page: each `arm_board` protobuffer explained.

The protobuffers for the `arm_board` are passed between software, control and embedded.

Movement_software_target.proto

Software -> control

This file contains two protobuffers detailing the source of a movement.

ArmBoardTargetMovement contains the target coordinates for a movement. This gets sent to control, they use this information to calculate motor angles.

ArmBoardObstructions is a placeholder for now. The idea is that software would be able to pass a list of coordinates of physical obstructions to control, around which they would manoeuvre the movement of the arm.

Movement_control_in.pb.h

Control -> Embedded

After control calculates the angles using the information from the previous protobuf, it sends the control signals back to us (embedded) with the **absolute** angles and frequency for the motors. **So, this is the information embedded uses to actuate the motors.**

ArmboardControlSignals contains all the angles the motors need to be turned to. We then use this information to set the PWM pins.

Movement_software_feedback.proto

Embedded -> Software

After the movement happens (or fails), embedded will send feedback to software for them to calculate the next movement.

ArmBoardMovementFeedback sends an error code to software.

Possible error codes are:

- Point_not_in_range
- Obstruction
- Calibration
- Motor_malfunction
- All_ok

ArmBoardActualPositions sends back the angle of each motor of the arm. Software will use this to display a 3D model of the arm position. (Allegedly)

Motor_diagnostics.proto

Arm board -> Debugging board

This protobuf gets periodically sent to the debugging board to indicate the status.

ArmBoardDiagnostics the `MotorInformation` for each motor and the state of the **entire** board.

Possible states are:

- Idle
- Operating
- Calibrating
- Errored

MotorInformation is a common protobuf shared between all boards that use motors (this and driving board). `MotorInformation` contains the state (same as above), `motor_id`, `rpm`, `voltage` and `encoder_angle` for a single motor.

Revision #4

Created 2026-04-14 08:58:41 UTC by Lisa te Braak

Updated 2026-04-21 15:55:49 UTC by Lisa te Braak