MAVLink Include Files: minimal.xml

MAVLink Protocol Version

The current MAVLink version is 2.3. The minor version numbers (after the dot) range from 1-255.

This file has protocol dialect: 0.

MAVLink Type Enumerations

FIRMWARE_VERSION_TYPE

[Enum] These values define the type of firmware release. These values indicate the first version or release of this type. For example the first alpha release would be 64, the second would be 65.

Value	Field Name	Description
0	FIRMWARE_VERSION_TYPE_DEV	development release
64	FIRMWARE_VERSION_TYPE_ALPHA	alpha release
128	FIRMWARE_VERSION_TYPE_BETA	beta release
192	FIRMWARE_VERSION_TYPE_RC	release candidate
255	FIRMWARE_VERSION_TYPE_OFFICIAL	official stable release

HL_FAILURE_FLAG

[Enum] Flags to report failure cases over the high latency telemtry.

Value	Field Name	Description
1	HL_FAILURE_FLAG_GPS	GPS failure.
2	HL_FAILURE_FLAG_DIFFERENTIAL_PRESSURE	Differential pressure sensor failure.
4	HL_FAILURE_FLAG_ABSOLUTE_PRESSURE	Absolute pressure sensor failure.
8	HL_FAILURE_FLAG_3D_ACCEL	Accelerometer sensor failure.
16	HL_FAILURE_FLAG_3D_GYRO	Gyroscope sensor failure.
32	HL_FAILURE_FLAG_3D_MAG	Magnetometer sensor failure.
64	HL_FAILURE_FLAG_TERRAIN	Terrain subsystem failure.
128	HL_FAILURE_FLAG_BATTERY	Battery failure/critical low battery.
256	HL_FAILURE_FLAG_RC_RECEIVER	RC receiver failure/no rc connection.
512	HL_FAILURE_FLAG_OFFBOARD_LINK	Offboard link failure.
1024	HL_FAILURE_FLAG_ENGINE	Engine failure.
2048	HL_FAILURE_FLAG_GEOFENCE	Geofence violation.
4096	HL_FAILURE_FLAG_ESTIMATOR	Estimator failure, for example measurement rejection or large variances.
8192	HL_FAILURE_FLAG_MISSION	Mission failure.

MAV_GOTO

[Enum] Actions that may be specified in MAV_CMD_OVERRIDE_GOTO to override mission execution.

Value	Field Name	Description
0	MAV_GOTO_DO_HOLD	Hold at the current position.
1	MAV_GOTO_DO_CONTINUE	Continue with the next item in mission execution.
2	MAV_GOTO_HOLD_AT_CURRENT_POSITION	Hold at the current position of the system
3	MAV_GOTO_HOLD_AT_SPECIFIED_POSITION	Hold at the position specified in the parameters of the DO_HOLD action

MAV_MODE

[Enum] These defines are predefined OR-combined mode flags. There is no need to use values from this enum, but it simplifies the use of the mode flags. Note that manual input is enabled in all modes as a safety override.

Value	Field Name	Description
0	MAV_MODE_PREFLIGHT	System is not ready to fly, booting, calibrating, etc. No flag is set.
80	MAV_MODE_STABILIZE_DISARMED	System is allowed to be active, under assisted RC control.
208	MAV_MODE_STABILIZE_ARMED	System is allowed to be active, under assisted RC control.
64	MAV_MODE_MANUAL_DISARMED	System is allowed to be active, under manual (RC) control, no stabilization
192	MAV_MODE_MANUAL_ARMED	System is allowed to be active, under manual (RC) control, no stabilization
88	MAV_MODE_GUIDED_DISARMED	System is allowed to be active, under autonomous control, manual setpoint
216	MAV_MODE_GUIDED_ARMED	System is allowed to be active, under autonomous control, manual setpoint
92	MAV_MODE_AUTO_DISARMED	System is allowed to be active, under autonomous control and navigation (the trajectory is decided onboard and not pre-programmed by waypoints)
220	MAV_MODE_AUTO_ARMED	System is allowed to be active, under autonomous control and navigation (the trajectory is decided onboard and not pre-programmed by waypoints)
66	MAV_MODE_TEST_DISARMED	UNDEFINED mode. This solely depends on the autopilot - use with caution, intended for developers only.
194	MAV_MODE_TEST_ARMED	UNDEFINED mode. This solely depends on the autopilot - use with caution, intended for developers only.

MAV_SYS_STATUS_SENSOR

[Enum] These encode the sensors whose status is sent as part of the SYS_STATUS message.

Value	Field Name	Description
1	MAV_SYS_STATUS_SENSOR_3D_GYRO	0x01 3D gyro
2	MAV_SYS_STATUS_SENSOR_3D_ACCEL	0x02 3D accelerometer
4	MAV_SYS_STATUS_SENSOR_3D_MAG	0x04 3D magnetometer
8	MAV_SYS_STATUS_SENSOR_ABSOLUTE_PRESSURE	0x08 absolute pressure
16	MAV_SYS_STATUS_SENSOR_DIFFERENTIAL_PRESSURE	0x10 differential pressure

32	MAV_SYS_STATUS_SENSOR_GPS	0x20 GPS
64	MAV_SYS_STATUS_SENSOR_OPTICAL_FLOW	0x40 optical flow
128	MAV_SYS_STATUS_SENSOR_VISION_POSITION	0x80 computer vision position
256	MAV_SYS_STATUS_SENSOR_LASER_POSITION	0x100 laser based position
512	MAV_SYS_STATUS_SENSOR_EXTERNAL_GROUND_TRUTH	0x200 external ground truth (Vicon or Leica)
1024	MAV_SYS_STATUS_SENSOR_ANGULAR_RATE_CONTROL	0x400 3D angular rate control
2048	MAV_SYS_STATUS_SENSOR_ATTITUDE_STABILIZATION	0x800 attitude stabilization
4096	MAV_SYS_STATUS_SENSOR_YAW_POSITION	0x1000 yaw position
8192	MAV_SYS_STATUS_SENSOR_Z_ALTITUDE_CONTROL	0x2000 z/altitude control
16384	MAV_SYS_STATUS_SENSOR_XY_POSITION_CONTROL	0x4000 x/y position control
32768	MAV_SYS_STATUS_SENSOR_MOTOR_OUTPUTS	0x8000 motor outputs / control
65536	MAV_SYS_STATUS_SENSOR_RC_RECEIVER	0x10000 rc receiver
131072	MAV_SYS_STATUS_SENSOR_3D_GYRO2	0x20000 2nd 3D gyro
262144	MAV_SYS_STATUS_SENSOR_3D_ACCEL2	0x40000 2nd 3D accelerometer
524288	MAV_SYS_STATUS_SENSOR_3D_MAG2	0x80000 2nd 3D magnetometer
1048576	MAV_SYS_STATUS_GEOFENCE	0x100000 geofence
2097152	MAV_SYS_STATUS_AHRS	0x200000 AHRS subsystem health
4194304	MAV_SYS_STATUS_TERRAIN	0x400000 Terrain subsystem health
8388608	MAV_SYS_STATUS_REVERSE_MOTOR	0x800000 Motors are reversed
16777216	MAV_SYS_STATUS_LOGGING	0x1000000 Logging
33554432	MAV_SYS_STATUS_SENSOR_BATTERY	0x2000000 Battery
67108864	MAV_SYS_STATUS_SENSOR_PROXIMITY	0x4000000 Proximity
134217728	MAV_SYS_STATUS_SENSOR_SATCOM	0x8000000 Satellite Communication
268435456	MAV_SYS_STATUS_PREARM_CHECK	0x10000000 pre-arm check status. Always healthy when armed
536870912	MAV_SYS_STATUS_OBSTACLE_AVOIDANCE	0x20000000 Avoidance/collision prevention
1073741824	MAV_SYS_STATUS_SENSOR_PROPULSION	0x40000000 propulsion (actuator, esc, motor or propellor)

MAV_SYS_STATUS_SENSOR_EXTENDED

[Enum] These encode the sensors whose status is sent as part of the SYS_STATUS message in the extended fields.

Value	Field Name	Description
1	MAV_SYS_STATUS_RECOVERY_SYSTEM	0x01 Recovery system (parachute, balloon, retracts etc)

MAV_FRAME

[Enum] Coordinate frames used by MAVLink. Not all frames are supported by all commands, messages, or vehicles.

Global frames use the following naming conventions:

- "GLOBAL": Global coordinate frame with WGS84 latitude/longitude and altitude positive over mean sea level (MSL) by default. The following modifiers may be used with "GLOBAL":
- "RELATIVE_ALT": Altitude is relative to the vehicle home position rather than MSL.
- "TERRAIN_ALT": Altitude is relative to ground level rather than MSL.
- "INT": Latitude/longitude (in degrees) are scaled by multiplying by 1E7.

Local frames use the following naming conventions:

- "LOCAL": Origin of local frame is fixed relative to earth. Unless otherwise specified this origin is the origin of the vehicle position-estimator ("EKF").
- "BODY": Origin of local frame travels with the vehicle. NOTE, "BODY" does NOT indicate alignment of frame axis with vehicle attitude.
- "OFFSET": Deprecated synonym for "BODY" (origin travels with the vehicle). Not to be used for new frames.

Some deprecated frames do not follow these conventions (e.g. MAV_FRAME_BODY_NED and MAV_FRAME_BODY_OFFSET_NED).

Value	Field Name	Description
0	MAV_FRAME_GLOBAL	Global (WGS84) coordinate frame + MSL altitude. First value / x: latitude, second value / y: longitude, third value / z: positive altitude over mean sea level (MSL).
1	MAV_FRAME_LOCAL_NED	NED local tangent frame (x: North, y: East, z: Down) with origin fixed relative to earth.
2	MAV_FRAME_MISSION	NOT a coordinate frame, indicates a mission command.
3	MAV_FRAME_GLOBAL_RELATIVE_ALT	Global (WGS84) coordinate frame + altitude relative to the home position. First value / x: latitude, second value / y: longitude, third value / z: positive altitude with 0 being at the altitude of the home position.
4	MAV_FRAME_LOCAL_ENU	ENU local tangent frame (x: East, y: North, z: Up) with origin fixed relative to earth.
5	MAV_FRAME_GLOBAL_INT	Global (WGS84) coordinate frame (scaled) + MSL altitude. First value / x: latitude in degrees*1E7, second value / y: longitude in degrees*1E7, third value / z: positive altitude over mean sea level (MSL).
6	MAV_FRAME_GLOBAL_RELATIVE_ALT_INT	Global (WGS84) coordinate frame (scaled) + altitude relative to the home position. First value / x: latitude in degrees*1E7, second value / y: longitude in degrees*1E7, third value / z: positive

altitude with 0 being at the altitude of the home position.		
7	MAV_FRAME_LOCAL_OFFSET_NED	NED local tangent frame (x: North, y: East, z: Down) with origin that travels with the vehicle.
8	MAV_FRAME_BODY_NED	

DEPRECATED: Replaced by MAV_FRAME_BODY_FRD (2019-08). | Same as MAV_FRAME_LOCAL_NED when used to represent position values. Same as MAV_FRAME_BODY_FRD when used with velocity/acceleration values. | | 9 | MAV_FRAME_BODY_OFFSET_NED

DEPRECATED: Replaced by MAV_FRAME_BODY_FRD (2019-08). | This is the same as MAV_FRAME_BODY_FRD. | | 10 | MAV_FRAME_GLOBAL_TERRAIN_ALT | Global (WGS84) coordinate frame with AGL altitude (at the waypoint coordinate). First value / x: latitude in degrees, second value / y: longitude in degrees, third value / z: positive altitude in meters with 0 being at ground level in terrain model. | | 11 | MAV_FRAME_GLOBAL_TERRAIN_ALT_INT | Global (WGS84) coordinate frame (scaled) with AGL altitude (at the waypoint coordinate). First value / x: latitude in degrees*1E7, second value / y: longitude in degrees*1E7, third value / z: positive altitude in meters with 0 being at ground level in terrain model. | | 12 | MAV_FRAME_BODY_FRD | FRD local tangent frame (x: Forward, y: Right, z: Down) with origin that travels with vehicle. The forward axis is aligned to the front of the vehicle in the horizontal plane. | | 13 | MAV_FRAME_RESERVED_13

DEPRECATED: Replaced by (2019-04). | MAV_FRAME_BODY_FLU - Body fixed frame of reference, Z-up (x: Forward, y: Left, z: Up). | | 14 | MAV_FRAME_RESERVED_14

DEPRECATED: Replaced by MAV_FRAME_LOCAL_FRD (2019-04). | MAV_FRAME_MOCAP_NED - Odometry local coordinate frame of data given by a motion capture system, Z-down (x: North, y: East, z: Down). | | 15 | MAV_FRAME_RESERVED_15

DEPRECATED: Replaced by MAV_FRAME_LOCAL_FLU (2019-04). | MAV_FRAME_MOCAP_ENU - Odometry local coordinate frame of data given by a motion capture system, Z-up (x: East, y: North, z: Up). | | 16 | MAV_FRAME_RESERVED_16

DEPRECATED: Replaced by MAV_FRAME_LOCAL_FRD (2019-04). | MAV_FRAME_VISION_NED - Odometry local coordinate frame of data given by a vision estimation system, Z-down (x: North, y: East, z: Down). | | 17 | MAV_FRAME_RESERVED_17

DEPRECATED: Replaced by MAV_FRAME_LOCAL_FLU (2019-04). | MAV_FRAME_VISION_ENU - Odometry local coordinate frame of data given by a vision estimation system, Z-up (x: East, y: North, z: Up). | | 18 | MAV_FRAME_RESERVED_18

DEPRECATED: Replaced by MAV_FRAME_LOCAL_FRD (2019-04). | MAV_FRAME_ESTIM_NED - Odometry local coordinate frame of data given by an estimator running onboard the vehicle, Z-down (x: North, y: East, z: Down). | | 19 | MAV_FRAME_RESERVED_19

DEPRECATED: Replaced by MAV_FRAME_LOCAL_FLU (2019-04). | MAV_FRAME_ESTIM_ENU - Odometry local coordinate frame of data given by an estimator running onboard the vehicle, Z-up (x: East, y: North, z: Up). | | 20 | MAV_FRAME_LOCAL_FRD | FRD local tangent frame (x: Forward, y: Right, z: Down) with origin fixed relative to earth. The forward axis is aligned to the front of the vehicle in the horizontal plane. | | 21 | MAV_FRAME_LOCAL_FLU | FLU local tangent frame (x: Forward, y: Left, z: Up) with origin fixed relative to earth. The forward axis is aligned to the front of the vehicle in the horizontal plane. |

MAVLINK_DATA_STREAM_TYPE

[Enum]

Value	Field Name	Description
0	MAVLINK_DATA_STREAM_IMG_JPEG	
1	MAVLINK_DATA_STREAM_IMG_BMP	
2	MAVLINK_DATA_STREAM_IMG_RAW8U	
3	MAVLINK_DATA_STREAM_IMG_RAW32U	
4	MAVLINK_DATA_STREAM_IMG_PGM	
5	MAVLINK_DATA_STREAM_IMG_PNG	

FENCE_ACTION

[Enum] Actions following geofence breach.

Value	Field Name	Description
0	FENCE_ACTION_NONE	Disable fenced mode. If used in a plan this would mean the next fence is disabled.
1	FENCE_ACTION_GUIDED	Fly to geofence MAV_CMD_NAV_FENCE_RETURN_POINT in GUIDED mode. Note: This action is only supported by ArduPlane, and may not be supported in all versions.
2	FENCE_ACTION_REPORT	Report fence breach, but don't take action
3	FENCE_ACTION_GUIDED_THR_PASS	Fly to geofence MAV_CMD_NAV_FENCE_RETURN_POINT with manual throttle control in GUIDED mode. Note: This action is only supported by ArduPlane, and may not be supported in all versions.
4	FENCE_ACTION_RTL	Return/RTL mode.
5	FENCE_ACTION_HOLD	Hold at current location.
6	FENCE_ACTION_TERMINATE	Termination failsafe. Motors are shut down (some flight stacks may trigger other failsafe actions).
7	FENCE_ACTION_LAND	Land at current location.

FENCE_BREACH

[Enum]

Value	Field Name	Description
0	FENCE_BREACH_NONE	No last fence breach
1	FENCE_BREACH_MINALT	Breached minimum altitude
2	FENCE_BREACH_MAXALT	Breached maximum altitude
3	FENCE_BREACH_BOUNDARY	Breached fence boundary

FENCE_MITIGATE

[Enum] Actions being taken to mitigate/prevent fence breach

Value	Field Name	Description
0	FENCE_MITIGATE_UNKNOWN	Unknown
1	FENCE_MITIGATE_NONE	No actions being taken

2	FENCE_MITIGATE_VEL_LIMIT	Velocity limiting active to prevent breach

MAV_MOUNT_MODE

DEPRECATED: Replaced by GIMBAL_MANAGER_FLAGS (2020-01).

[Enum] Enumeration of possible mount operation modes. This message is used by obsolete/deprecated gimbal messages.

Value	Field Name	Description
0	MAV_MOUNT_MODE_RETRACT	Load and keep safe position (Roll,Pitch,Yaw) from permant memory and stop stabilization
1	MAV_MOUNT_MODE_NEUTRAL	Load and keep neutral position (Roll,Pitch,Yaw) from permanent memory.
2	MAV_MOUNT_MODE_MAVLINK_TARGETING	Load neutral position and start MAVLink Roll,Pitch,Yaw control with stabilization
3	MAV_MOUNT_MODE_RC_TARGETING	Load neutral position and start RC Roll,Pitch,Yaw control with stabilization
4	MAV_MOUNT_MODE_GPS_POINT	Load neutral position and start to point to Lat,Lon,Alt
5	MAV_MOUNT_MODE_SYSID_TARGET	Gimbal tracks system with specified system ID
6	MAV_MOUNT_MODE_HOME_LOCATION	Gimbal tracks home position

GIMBAL_DEVICE_CAP_FLAGS

[Enum] Gimbal device (low level) capability flags (bitmap)

Value	Field Name	Description
1	GIMBAL_DEVICE_CAP_FLAGS_HAS_RETRACT	Gimbal device supports a retracted position
2	GIMBAL_DEVICE_CAP_FLAGS_HAS_NEUTRAL	Gimbal device supports a horizontal, forward looking position, stabilized
4	GIMBAL_DEVICE_CAP_FLAGS_HAS_ROLL_AXIS	Gimbal device supports rotating around roll axis.
8	GIMBAL_DEVICE_CAP_FLAGS_HAS_ROLL_FOLLOW	Gimbal device supports to follow a roll angle relative to the vehicle
16	GIMBAL_DEVICE_CAP_FLAGS_HAS_ROLL_LOCK	Gimbal device supports locking to an roll angle (generally that's the default with roll stabilized)
32	GIMBAL_DEVICE_CAP_FLAGS_HAS_PITCH_AXIS	Gimbal device supports rotating around pitch axis.
64	GIMBAL_DEVICE_CAP_FLAGS_HAS_PITCH_FOLLOW	Gimbal device supports to follow a pitch angle relative to the vehicle
128	GIMBAL_DEVICE_CAP_FLAGS_HAS_PITCH_LOCK	Gimbal device supports locking to an pitch angle (generally that's the default with pitch stabilized)
256	GIMBAL_DEVICE_CAP_FLAGS_HAS_YAW_AXIS	Gimbal device supports rotating around yaw axis.
512	GIMBAL_DEVICE_CAP_FLAGS_HAS_YAW_FOLLOW	Gimbal device supports to follow a yaw angle relative to the vehicle (generally that's the default)
1024	GIMBAL_DEVICE_CAP_FLAGS_HAS_YAW_LOCK	Gimbal device supports locking to an absolute heading (often this is an option available)
2048	GIMBAL_DEVICE_CAP_FLAGS_SUPPORTS_INFINITE_YAW	Gimbal device supports yawing/panning infinetely (e.g. using slip disk).

GIMBAL_MANAGER_CAP_FLAGS

[Enum] Gimbal manager high level capability flags (bitmap). The first 16 bits are identical to the GIMBAL_DEVICE_CAP_FLAGS. However, the gimbal manager does not need to copy the flags from the gimbal but can also enhance the capabilities and thus add flags.

Value	Field Name	Description
1	GIMBAL_MANAGER_CAP_FLAGS_HAS_RETRACT	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_RETRACT.
2	GIMBAL_MANAGER_CAP_FLAGS_HAS_NEUTRAL	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_NEUTRAL.
4	GIMBAL_MANAGER_CAP_FLAGS_HAS_ROLL_AXIS	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_ROLL_AXIS.
8	GIMBAL_MANAGER_CAP_FLAGS_HAS_ROLL_FOLLOW	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_ROLL_FOLLOW.
16	GIMBAL_MANAGER_CAP_FLAGS_HAS_ROLL_LOCK	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_ROLL_LOCK.
32	GIMBAL_MANAGER_CAP_FLAGS_HAS_PITCH_AXIS	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_PITCH_AXIS.
64	GIMBAL_MANAGER_CAP_FLAGS_HAS_PITCH_FOLLOW	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_PITCH_FOLLOW.
128	GIMBAL_MANAGER_CAP_FLAGS_HAS_PITCH_LOCK	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_PITCH_LOCK.
256	GIMBAL_MANAGER_CAP_FLAGS_HAS_YAW_AXIS	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_YAW_AXIS.
512	GIMBAL_MANAGER_CAP_FLAGS_HAS_YAW_FOLLOW	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_YAW_FOLLOW.
1024	GIMBAL_MANAGER_CAP_FLAGS_HAS_YAW_LOCK	Based on GIMBAL_DEVICE_CAP_FLAGS_HAS_YAW_LOCK.
2048	GIMBAL_MANAGER_CAP_FLAGS_SUPPORTS_INFINITE_YAW	Based on GIMBAL_DEVICE_CAP_FLAGS_SUPPORTS_INFINITE_YAW.
65536	GIMBAL_MANAGER_CAP_FLAGS_CAN_POINT_LOCATION_LOCAL	Gimbal manager supports to point to a local position.
131072	GIMBAL_MANAGER_CAP_FLAGS_CAN_POINT_LOCATION_GLOBAL	Gimbal manager supports to point to a global latitude, longitude, altitude position.

GIMBAL_DEVICE_FLAGS

[Enum] Flags for gimbal device (lower level) operation.

Value	Field Name	Description
1	GIMBAL_DEVICE_FLAGS_RETRACT	Set to retracted safe position (no stabilization), takes presedence over all other flags.
2	GIMBAL_DEVICE_FLAGS_NEUTRAL	Set to neutral/default position, taking precedence over all other flags except RETRACT. Neutral is commonly forward-facing and horizontal (pitch=yaw=0) but may be any orientation.
4	GIMBAL_DEVICE_FLAGS_ROLL_LOCK	Lock roll angle to absolute angle relative to horizon (not relative to drone). This is generally the default with a stabilizing gimbal.
8	GIMBAL_DEVICE_FLAGS_PITCH_LOCK	Lock pitch angle to absolute angle relative to horizon (not relative to drone). This is generally the default.
16	GIMBAL_DEVICE_FLAGS_YAW_LOCK	Lock yaw angle to absolute angle relative to North (not relative to drone). If this flag is set, the quaternion is in the Earth frame with the x-axis pointing North (yaw absolute). If this flag is not set, the quaternion frame is in the Earth frame rotated so that the x-axis is pointing forward (yaw relative to vehicle).

GIMBAL_MANAGER_FLAGS

[Enum] Flags for high level gimbal manager operation The first 16 bits are identical to the GIMBAL_DEVICE_FLAGS.

Value	Field Name	Description
1	GIMBAL_MANAGER_FLAGS_RETRACT	Based on GIMBAL_DEVICE_FLAGS_RETRACT
2	GIMBAL_MANAGER_FLAGS_NEUTRAL	Based on GIMBAL_DEVICE_FLAGS_NEUTRAL
4	GIMBAL_MANAGER_FLAGS_ROLL_LOCK	Based on GIMBAL_DEVICE_FLAGS_ROLL_LOCK
8	GIMBAL_MANAGER_FLAGS_PITCH_LOCK	Based on GIMBAL_DEVICE_FLAGS_PITCH_LOCK
16	GIMBAL_MANAGER_FLAGS_YAW_LOCK	Based on GIMBAL_DEVICE_FLAGS_YAW_LOCK

GIMBAL_DEVICE_ERROR_FLAGS

[Enum] Gimbal device (low level) error flags (bitmap, 0 means no error)

Value	Field Name	Description
1	GIMBAL_DEVICE_ERROR_FLAGS_AT_ROLL_LIMIT	Gimbal device is limited by hardware roll limit.
2	GIMBAL_DEVICE_ERROR_FLAGS_AT_PITCH_LIMIT	Gimbal device is limited by hardware pitch limit.
4	GIMBAL_DEVICE_ERROR_FLAGS_AT_YAW_LIMIT	Gimbal device is limited by hardware yaw limit.
8	GIMBAL_DEVICE_ERROR_FLAGS_ENCODER_ERROR	There is an error with the gimbal encoders.
16	GIMBAL_DEVICE_ERROR_FLAGS_POWER_ERROR	There is an error with the gimbal power source.
32	GIMBAL_DEVICE_ERROR_FLAGS_MOTOR_ERROR	There is an error with the gimbal motor's.
64	GIMBAL_DEVICE_ERROR_FLAGS_SOFTWARE_ERROR	There is an error with the gimbal's software.
128	GIMBAL_DEVICE_ERROR_FLAGS_COMMS_ERROR	There is an error with the gimbal's communication.
256	GIMBAL_DEVICE_ERROR_FLAGS_CALIBRATION_RUNNING	Gimbal is currently calibrating.

GRIPPER_ACTIONS

[Enum] Gripper actions.

Value	Field Name	Description
0	GRIPPER_ACTION_RELEASE	Gripper release cargo.
1	GRIPPER_ACTION_GRAB	Gripper grab onto cargo.

WINCH_ACTIONS

[Enum] Winch actions.

Value	Field Name	Description
0	WINCH_RELAXED	Allow motor to freewheel.
1	WINCH_RELATIVE_LENGTH_CONTROL	Wind or unwind specified length of line, optionally using specified rate.
2	WINCH_RATE_CONTROL	Wind or unwind line at specified rate.
3	WINCH_LOCK	Perform the locking sequence to relieve motor while in the fully retracted position. Only action and instance command parameters are used, others are ignored.
4	WINCH_DELIVER	Sequence of drop, slow down, touch down, reel up, lock. Only action and instance command parameters are used, others are ignored.
5	WINCH_HOLD	Engage motor and hold current position. Only action and instance command parameters are used, others are ignored.
6	WINCH_RETRACT	Return the reel to the fully retracted position. Only action and instance command parameters are used, others are ignored.
7	WINCH_LOAD_LINE	Load the reel with line. The winch will calculate the total loaded length and stop when the tension exceeds a threshold. Only action and instance command parameters are used, others are ignored.
8	WINCH_ABANDON_LINE	Spool out the entire length of the line. Only action and instance command parameters are used, others are ignored.

UAVCAN_NODE_HEALTH

[Enum] Generalized UAVCAN node health

Value	Field Name	Description
0	UAVCAN_NODE_HEALTH_OK	The node is functioning properly.
1	UAVCAN_NODE_HEALTH_WARNING	A critical parameter went out of range or the node has encountered a minor failure.
2	UAVCAN_NODE_HEALTH_ERROR	The node has encountered a major failure.
3	UAVCAN_NODE_HEALTH_CRITICAL	The node has suffered a fatal malfunction.

UAVCAN_NODE_MODE

[Enum] Generalized UAVCAN node mode

Value	Field Name	Description
0	UAVCAN_NODE_MODE_OPERATIONAL	The node is performing its primary functions.
1	UAVCAN_NODE_MODE_INITIALIZATION	The node is initializing; this mode is entered immediately after startup.
2	UAVCAN_NODE_MODE_MAINTENANCE	The node is under maintenance.
3	UAVCAN_NODE_MODE_SOFTWARE_UPDATE	The node is in the process of updating its software.
7	UAVCAN_NODE_MODE_OFFLINE	The node is no longer available online.

ESC_CONNECTION_TYPE

[Enum] Indicates the ESC connection type.

Value	Field Name	Description
0	ESC_CONNECTION_TYPE_PPM	Traditional PPM ESC.
1	ESC_CONNECTION_TYPE_SERIAL	Serial Bus connected ESC.
2	ESC_CONNECTION_TYPE_ONESHOT	One Shot PPM ESC.

3	ESC_CONNECTION_TYPE_I2C	I2C ESC.
4	ESC_CONNECTION_TYPE_CAN	CAN-Bus ESC.
5	ESC_CONNECTION_TYPE_DSHOT	DShot ESC.

ESC_FAILURE_FLAGS

[Enum] Flags to report ESC failures.

Value	Field Name	Description
0	ESC_FAILURE_NONE	No ESC failure.
1	ESC_FAILURE_OVER_CURRENT	Over current failure.
2	ESC_FAILURE_OVER_VOLTAGE	Over voltage failure.
4	ESC_FAILURE_OVER_TEMPERATURE	Over temperature failure.
8	ESC_FAILURE_OVER_RPM	Over RPM failure.
16	ESC_FAILURE_INCONSISTENT_CMD	Inconsistent command failure i.e. out of bounds.
32	ESC_FAILURE_MOTOR_STUCK	Motor stuck failure.
64	ESC_FAILURE_GENERIC	Generic ESC failure.

STORAGE_STATUS

[Enum] Flags to indicate the status of camera storage.

Value	Field Name	Description
0	STORAGE_STATUS_EMPTY	Storage is missing (no microSD card loaded for example.)
1	STORAGE_STATUS_UNFORMATTED	Storage present but unformatted.
2	STORAGE_STATUS_READY	Storage present and ready.
3	STORAGE_STATUS_NOT_SUPPORTED	Camera does not supply storage status information. Capacity information in STORAGE_INFORMATION fields will be ignored.

STORAGE_TYPE

[Enum] Flags to indicate the type of storage.

Value	Field Name	Description
0	STORAGE_TYPE_UNKNOWN	Storage type is not known.
1	STORAGE_TYPE_USB_STICK	Storage type is USB device.
2	STORAGE_TYPE_SD	Storage type is SD card.
3	STORAGE_TYPE_MICROSD	Storage type is microSD card.
4	STORAGE_TYPE_CF	Storage type is CFast.
5	STORAGE_TYPE_CFE	Storage type is CFexpress.
6	STORAGE_TYPE_XQD	Storage type is XQD.
7	STORAGE_TYPE_HD	Storage type is HD mass storage type.
254	STORAGE_TYPE_OTHER	Storage type is other, not listed type.

STORAGE_USAGE_FLAG

[Enum] Flags to indicate usage for a particular storage (see STORAGE_INFORMATION.storage_usage and MAV_CMD_SET_STORAGE_USAGE).

Value	Field Name	Description
1	STORAGE_USAGE_FLAG_SET	Always set to 1 (indicates STORAGE_INFORMATION.storage_usage is supported).
2	STORAGE_USAGE_FLAG_PHOTO	Storage for saving photos.
4	STORAGE_USAGE_FLAG_VIDEO	Storage for saving videos.
8	STORAGE_USAGE_FLAG_LOGS	Storage for saving logs.

ORBIT_YAW_BEHAVIOUR

[Enum] Yaw behaviour during orbit flight.

Value	Field Name	Description
0	ORBIT_YAW_BEHAVIOUR_HOLD_FRONT_TO_CIRCLE_CENTER	Vehicle front points to the center (default).
1	ORBIT_YAW_BEHAVIOUR_HOLD_INITIAL_HEADING	Vehicle front holds heading when message received.
2	ORBIT_YAW_BEHAVIOUR_UNCONTROLLED	Yaw uncontrolled.
3	ORBIT_YAW_BEHAVIOUR_HOLD_FRONT_TANGENT_TO_CIRCLE	Vehicle front follows flight path (tangential to circle).
4	ORBIT_YAW_BEHAVIOUR_RC_CONTROLLED	Yaw controlled by RC input.

WIFI_CONFIG_AP_RESPONSE

[Enum] Possible responses from a WIFI_CONFIG_AP message.

Value	Field Name	Description
0	WIFI_CONFIG_AP_RESPONSE_UNDEFINED	Undefined response. Likely an indicative of a system that doesn't support this request.
1	WIFI_CONFIG_AP_RESPONSE_ACCEPTED	Changes accepted.
2	WIFI_CONFIG_AP_RESPONSE_REJECTED	Changes rejected.
3	WIFI_CONFIG_AP_RESPONSE_MODE_ERROR	Invalid Mode.
4	WIFI_CONFIG_AP_RESPONSE_SSID_ERROR	Invalid SSID.
5	WIFI_CONFIG_AP_RESPONSE_PASSWORD_ERROR	Invalid Password.

CELLULAR_CONFIG_RESPONSE

[Enum] Possible responses from a CELLULAR_CONFIG message.

Value	Field Name	Description
0	CELLULAR_CONFIG_RESPONSE_ACCEPTED	Changes accepted.

1	CELLULAR_CONFIG_RESPONSE_APN_ERROR	Invalid APN.
2	CELLULAR_CONFIG_RESPONSE_PIN_ERROR	Invalid PIN.
3	CELLULAR_CONFIG_RESPONSE_REJECTED	Changes rejected.
4	CELLULAR_CONFIG_BLOCKED_PUK_REQUIRED	PUK is required to unblock SIM card.

WIFI_CONFIG_AP_MODE

[Enum] WiFi Mode.

Value	Field Name	Description
0	WIFI_CONFIG_AP_MODE_UNDEFINED	WiFi mode is undefined.
1	WIFI_CONFIG_AP_MODE_AP	WiFi configured as an access point.
2	WIFI_CONFIG_AP_MODE_STATION	WiFi configured as a station connected to an existing local WiFi network.
3	WIFI_CONFIG_AP_MODE_DISABLED	WiFi disabled.

COMP_METADATA_TYPE

[Enum] Supported component metadata types. These are used in the "general" metadata file returned by COMPONENT_METADATA to provide information about supported metadata types. The types are not used directly in MAVLink messages.

Value	Field Name	Description
0	COMP_METADATA_TYPE_GENERAL	General information about the component. General metadata includes information about other metadata types supported by the component. Files of this type must be supported, and must be downloadable from vehicle using a MAVLink FTP URI.
1	COMP_METADATA_TYPE_PARAMETER	Parameter meta data.
2	COMP_METADATA_TYPE_COMMANDS	Meta data that specifies which commands and command parameters the vehicle supports. (WIP)
3	COMP_METADATA_TYPE_PERIPHERALS	Meta data that specifies external non-MAVLink peripherals.
4	COMP_METADATA_TYPE_EVENTS	Meta data for the events interface.
5	COMP_METADATA_TYPE_ACTUATORS	Meta data for actuator configuration (motors, servos and vehicle geometry) and testing.

ACTUATOR_CONFIGURATION

[Enum] Actuator configuration, used to change a setting on an actuator. Component information metadata can be used to know which outputs support which commands.

Value	Field Name	Description
0	ACTUATOR_CONFIGURATION_NONE	Do nothing.
1	ACTUATOR_CONFIGURATION_BEEP	Command the actuator to beep now.
2	ACTUATOR_CONFIGURATION_3D_MODE_ON	Permanently set the actuator (ESC) to 3D mode (reversible thrust).
3	ACTUATOR_CONFIGURATION_3D_MODE_OFF	Permanently set the actuator (ESC) to non 3D mode (non-reversible thrust).
4	ACTUATOR_CONFIGURATION_SPIN_DIRECTION1	Permanently set the actuator (ESC) to spin direction 1 (which can be clockwise or counter-clockwise).
5	ACTUATOR_CONFIGURATION_SPIN_DIRECTION2	Permanently set the actuator (ESC) to spin direction 2 (opposite of direction 1).

ACTUATOR_OUTPUT_FUNCTION

[Enum] Actuator output function. Values greater or equal to 1000 are autopilot-specific.

Value	Field Name	Description
0	ACTUATOR_OUTPUT_FUNCTION_NONE	No function (disabled).
1	ACTUATOR_OUTPUT_FUNCTION_MOTOR1	Motor 1
2	ACTUATOR_OUTPUT_FUNCTION_MOTOR2	Motor 2
3	ACTUATOR_OUTPUT_FUNCTION_MOTOR3	Motor 3
4	ACTUATOR_OUTPUT_FUNCTION_MOTOR4	Motor 4
5	ACTUATOR_OUTPUT_FUNCTION_MOTOR5	Motor 5
6	ACTUATOR_OUTPUT_FUNCTION_MOTOR6	Motor 6
7	ACTUATOR_OUTPUT_FUNCTION_MOTOR7	Motor 7
8	ACTUATOR_OUTPUT_FUNCTION_MOTOR8	Motor 8
9	ACTUATOR_OUTPUT_FUNCTION_MOTOR9	Motor 9
10	ACTUATOR_OUTPUT_FUNCTION_MOTOR10	Motor 10
11	ACTUATOR_OUTPUT_FUNCTION_MOTOR11	Motor 11
12	ACTUATOR_OUTPUT_FUNCTION_MOTOR12	Motor 12
13	ACTUATOR_OUTPUT_FUNCTION_MOTOR13	Motor 13
14	ACTUATOR_OUTPUT_FUNCTION_MOTOR14	Motor 14
15	ACTUATOR_OUTPUT_FUNCTION_MOTOR15	Motor 15
16	ACTUATOR_OUTPUT_FUNCTION_MOTOR16	Motor 16
33	ACTUATOR_OUTPUT_FUNCTION_SERVO1	Servo 1
34	ACTUATOR_OUTPUT_FUNCTION_SERVO2	Servo 2
35	ACTUATOR_OUTPUT_FUNCTION_SERVO3	Servo 3
36	ACTUATOR_OUTPUT_FUNCTION_SERVO4	Servo 4
37	ACTUATOR_OUTPUT_FUNCTION_SERVO5	Servo 5
38	ACTUATOR_OUTPUT_FUNCTION_SERVO6	Servo 6
39	ACTUATOR_OUTPUT_FUNCTION_SERVO7	Servo 7
40	ACTUATOR_OUTPUT_FUNCTION_SERVO8	Servo 8
41	ACTUATOR_OUTPUT_FUNCTION_SERVO9	Servo 9
42	ACTUATOR_OUTPUT_FUNCTION_SERVO10	Servo 10
43	ACTUATOR_OUTPUT_FUNCTION_SERVO11	Servo 11
44	ACTUATOR_OUTPUT_FUNCTION_SERVO12	Servo 12
45	ACTUATOR_OUTPUT_FUNCTION_SERVO13	Servo 13
46	ACTUATOR_OUTPUT_FUNCTION_SERVO14	Servo 14
47	ACTUATOR_OUTPUT_FUNCTION_SERVO15	Servo 15

48	ACTUATOR_OUTPUT_FUNCTION_SERVO16	Servo 16

AUTOTUNE_AXIS

[Enum] Enable axes that will be tuned via autotuning. Used in MAV_CMD_DO_AUTOTUNE_ENABLE.

Value	Field Name	Description
0	AUTOTUNE_AXIS_DEFAULT	Flight stack tunes axis according to its default settings.
1	AUTOTUNE_AXIS_ROLL	Autotune roll axis.
2	AUTOTUNE_AXIS_PITCH	Autotune pitch axis.
4	AUTOTUNE_AXIS_YAW	Autotune yaw axis.

PREFLIGHT_STORAGE_PARAMETER_ACTION

[Enum] Actions for reading/writing parameters between persistent and volatile storage when using MAV_CMD_PREFLIGHT_STORAGE. (Commonly parameters are loaded from persistent storage (flash/EEPROM) into volatile storage (RAM) on startup and written back when they are changed.)

Value	Field Name	Description
0	PARAM_READ_PERSISTENT	Read all parameters from persistent storage. Replaces values in volatile storage.
1	PARAM_WRITE_PERSISTENT	Write all parameter values to persistent storage (flash/EEPROM)
2	PARAM_RESET_CONFIG_DEFAULT	Reset all user configurable parameters to their default value (including airframe selection, sensor calibration data, safety settings, and so on). Does not reset values that contain operation counters and vehicle computed statistics.
3	PARAM_RESET_SENSOR_DEFAULT	Reset only sensor calibration parameters to factory defaults (or firmware default if not available)
4	PARAM_RESET_ALL_DEFAULT	Reset all parameters, including operation counters, to default values

PREFLIGHT_STORAGE_MISSION_ACTION

[Enum] Actions for reading and writing plan information (mission, rally points, geofence) between persistent and volatile storage when using MAV_CMD_PREFLIGHT_STORAGE. (Commonly missions are loaded from persistent storage (flash/EEPROM) into volatile storage (RAM) on startup and written back when they are changed.)

Value	Field Name	Description
0	MISSION_READ_PERSISTENT	Read current mission data from persistent storage
1	MISSION_WRITE_PERSISTENT	Write current mission data to persistent storage
2	MISSION_RESET_DEFAULT	Erase all mission data stored on the vehicle (both persistent and volatile storage)

MAV_DATA_STREAM

DEPRECATED: Replaced by MESSAGE_INTERVAL (2015-06).

[Enum] A data stream is not a fixed set of messages, but rather a recommendation to the autopilot software. Individual autopilots may or may not obey the recommended messages.

Value	Field Name	Description
0	MAV_DATA_STREAM_ALL	Enable all data streams
1	MAV_DATA_STREAM_RAW_SENSORS	Enable IMU_RAW, GPS_RAW, GPS_STATUS packets.
2	MAV_DATA_STREAM_EXTENDED_STATUS	Enable GPS_STATUS, CONTROL_STATUS, AUX_STATUS
3	MAV_DATA_STREAM_RC_CHANNELS	Enable RC_CHANNELS_SCALED, RC_CHANNELS_RAW, SERVO_OUTPUT_RAW
4	MAV_DATA_STREAM_RAW_CONTROLLER	Enable ATTITUDE_CONTROLLER_OUTPUT, POSITION_CONTROLLER_OUTPUT, NAV_CONTROLLER_OUTPUT.
6	MAV_DATA_STREAM_POSITION	Enable LOCAL_POSITION, GLOBAL_POSITION_INT messages.
10	MAV_DATA_STREAM_EXTRA1	Dependent on the autopilot
11	MAV_DATA_STREAM_EXTRA2	Dependent on the autopilot
12	MAV_DATA_STREAM_EXTRA3	Dependent on the autopilot

MAV_ROI

DEPRECATED: Replaced by MAV_CMD_DO_SET_ROI_* (2018-01).

[Enum] The ROI (region of interest) for the vehicle. This can be be used by the vehicle for camera/vehicle attitude alignment (see MAV_CMD_NAV_ROI).

Value	Field Name	Description
0	MAV_ROI_NONE	No region of interest.
1	MAV_ROI_WPNEXT	Point toward next waypoint, with optional pitch/roll/yaw offset.
2	MAV_ROI_WPINDEX	Point toward given waypoint.
3	MAV_ROI_LOCATION	Point toward fixed location.
4	MAV_ROI_TARGET	Point toward of given id.

MAV_CMD_ACK

[Enum] ACK / NACK / ERROR values as a result of MAV_CMDs and for mission item transmission.

Value	Field Name	Description
0	MAV_CMD_ACK_OK	Command / mission item is ok.
1	MAV_CMD_ACK_ERR_FAIL	Generic error message if none of the other reasons fails or if no detailed error reporting is implemented.
2	MAV_CMD_ACK_ERR_ACCESS_DENIED	The system is refusing to accept this command from this source / communication partner.
3	MAV_CMD_ACK_ERR_NOT_SUPPORTED	Command or mission item is not supported, other commands would be accepted.
4	MAV_CMD_ACK_ERR_COORDINATE_FRAME_NOT_SUPPORTED	The coordinate frame of this command / mission item is not supported.

5	MAV_CMD_ACK_ERR_COORDINATES_OUT_OF_RANGE	The coordinate frame of this command is ok, but he coordinate values exceed the safety limits of this system. This is a generic error, please use the more specific error messages below if possible.
6	MAV_CMD_ACK_ERR_X_LAT_OUT_OF_RANGE	The X or latitude value is out of range.
7	MAV_CMD_ACK_ERR_Y_LON_OUT_OF_RANGE	The Y or longitude value is out of range.
8	MAV_CMD_ACK_ERR_Z_ALT_OUT_OF_RANGE	The Z or altitude value is out of range.

MAV_PARAM_TYPE

[Enum] Specifies the datatype of a MAVLink parameter.

Value	Field Name	Description
1	MAV_PARAM_TYPE_UINT8	8-bit unsigned integer
2	MAV_PARAM_TYPE_INT8	8-bit signed integer
3	MAV_PARAM_TYPE_UINT16	16-bit unsigned integer
4	MAV_PARAM_TYPE_INT16	16-bit signed integer
5	MAV_PARAM_TYPE_UINT32	32-bit unsigned integer
6	MAV_PARAM_TYPE_INT32	32-bit signed integer
7	MAV_PARAM_TYPE_UINT64	64-bit unsigned integer
8	MAV_PARAM_TYPE_INT64	64-bit signed integer
9	MAV_PARAM_TYPE_REAL32	32-bit floating-point
10	MAV_PARAM_TYPE_REAL64	64-bit floating-point

MAV_PARAM_EXT_TYPE

[Enum] Specifies the datatype of a MAVLink extended parameter.

Value	Field Name	Description
1	MAV_PARAM_EXT_TYPE_UINT8	8-bit unsigned integer
2	MAV_PARAM_EXT_TYPE_INT8	8-bit signed integer
3	MAV_PARAM_EXT_TYPE_UINT16	16-bit unsigned integer
4	MAV_PARAM_EXT_TYPE_INT16	16-bit signed integer
5	MAV_PARAM_EXT_TYPE_UINT32	32-bit unsigned integer
6	MAV_PARAM_EXT_TYPE_INT32	32-bit signed integer
7	MAV_PARAM_EXT_TYPE_UINT64	64-bit unsigned integer
8	MAV_PARAM_EXT_TYPE_INT64	64-bit signed integer
9	MAV_PARAM_EXT_TYPE_REAL32	32-bit floating-point
10	MAV_PARAM_EXT_TYPE_REAL64	64-bit floating-point
11	MAV_PARAM_EXT_TYPE_CUSTOM	Custom Type

MAV_RESULT

[Enum] Result from a MAVLink command (MAV_CMD)

Value	Field Name	Description
0	MAV_RESULT_ACCEPTED	Command is valid (is supported and has valid parameters), and was executed.
1	MAV_RESULT_TEMPORARILY_REJECTED	Command is valid, but cannot be executed at this time. This is used to indicate a problem that should be fixed just by waiting (e.g. a state machine is busy, can't arm because have not got GPS lock, etc.). Retrying later should work.
2	MAV_RESULT_DENIED	Command is invalid (is supported but has invalid parameters). Retrying same command and parameters will not work.
3	MAV_RESULT_UNSUPPORTED	Command is not supported (unknown).
4	MAV_RESULT_FAILED	Command is valid, but execution has failed. This is used to indicate any non-temporary or unexpected problem, i.e. any problem that must be fixed before the command can succeed/be retried. For example, attempting to write a file when out of memory, attempting to arm when sensors are not calibrated, etc.
5	MAV_RESULT_IN_PROGRESS	Command is valid and is being executed. This will be followed by further progress updates, i.e. the component may send further COMMAND_ACK messages with result MAV_RESULT_IN_PROGRESS (at a rate decided by the implementation), and must terminate by sending a COMMAND_ACK message with final result of the operation. The COMMAND_ACK.progress field can be used to indicate the progress of the operation.
6	MAV_RESULT_CANCELLED	Command has been cancelled (as a result of receiving a COMMAND_CANCEL message).

MAV_MISSION_RESULT

[Enum] Result of mission operation (in a MISSION_ACK message).

Value	Field Name	Description
0	MAV_MISSION_ACCEPTED	mission accepted OK
1	MAV_MISSION_ERROR	Generic error / not accepting mission commands at all right now.
2	MAV_MISSION_UNSUPPORTED_FRAME	Coordinate frame is not supported.
3	MAV_MISSION_UNSUPPORTED	Command is not supported.
4	MAV_MISSION_NO_SPACE	Mission items exceed storage space.
5	MAV_MISSION_INVALID	One of the parameters has an invalid value.
6	MAV_MISSION_INVALID_PARAM1	param1 has an invalid value.
7	MAV_MISSION_INVALID_PARAM2	param2 has an invalid value.
8	MAV_MISSION_INVALID_PARAM3	param3 has an invalid value.
9	MAV_MISSION_INVALID_PARAM4	param4 has an invalid value.
10	MAV_MISSION_INVALID_PARAM5_X	x / param5 has an invalid value.
11	MAV_MISSION_INVALID_PARAM6_Y	y / param6 has an invalid value.

12	MAV_MISSION_INVALID_PARAM7	z / param7 has an invalid value.
13	MAV_MISSION_INVALID_SEQUENCE	Mission item received out of sequence
14	MAV_MISSION_DENIED	Not accepting any mission commands from this communication partner.
15	MAV_MISSION_OPERATION_CANCELLED	Current mission operation cancelled (e.g. mission upload, mission download).

MAV_SEVERITY

[Enum] Indicates the severity level, generally used for status messages to indicate their relative urgency. Based on RFC-5424 using expanded definitions at: http://www.kiwisyslog.com/kb/info:-syslog-message-levels/.

Value	Field Name	Description
0	MAV_SEVERITY_EMERGENCY	System is unusable. This is a "panic" condition.
1	MAV_SEVERITY_ALERT	Action should be taken immediately. Indicates error in non-critical systems.
2	MAV_SEVERITY_CRITICAL	Action must be taken immediately. Indicates failure in a primary system.
3	MAV_SEVERITY_ERROR	Indicates an error in secondary/redundant systems.
4	MAV_SEVERITY_WARNING	Indicates about a possible future error if this is not resolved within a given timeframe. Example would be a low battery warning.
5	MAV_SEVERITY_NOTICE	An unusual event has occurred, though not an error condition. This should be investigated for the root cause.
6	MAV_SEVERITY_INFO	Normal operational messages. Useful for logging. No action is required for these messages.
7	MAV_SEVERITY_DEBUG	Useful non-operational messages that can assist in debugging. These should not occur during normal operation.

MAV_POWER_STATUS

[Enum] Power supply status flags (bitmask)

Value	Field Name	Description
1	MAV_POWER_STATUS_BRICK_VALID	main brick power supply valid
2	MAV_POWER_STATUS_SERVO_VALID	main servo power supply valid for FMU
4	MAV_POWER_STATUS_USB_CONNECTED	USB power is connected
8	MAV_POWER_STATUS_PERIPH_OVERCURRENT	peripheral supply is in over-current state
16	MAV_POWER_STATUS_PERIPH_HIPOWER_OVERCURRENT	hi-power peripheral supply is in over-current state
32	MAV_POWER_STATUS_CHANGED	Power status has changed since boot

SERIAL_CONTROL_DEV

[Enum] SERIAL_CONTROL device types

Value	Field Name	Description
0	SERIAL_CONTROL_DEV_TELEM1	First telemetry port
1	SERIAL_CONTROL_DEV_TELEM2	Second telemetry port
2	SERIAL_CONTROL_DEV_GPS1	First GPS port
3	SERIAL_CONTROL_DEV_GPS2	Second GPS port
10	SERIAL_CONTROL_DEV_SHELL	system shell
100	SERIAL_CONTROL_SERIAL0	SERIAL0
101	SERIAL_CONTROL_SERIAL1	SERIAL1
102	SERIAL_CONTROL_SERIAL2	SERIAL2
103	SERIAL_CONTROL_SERIAL3	SERIAL3
104	SERIAL_CONTROL_SERIAL4	SERIAL4
105	SERIAL_CONTROL_SERIAL5	SERIAL5
106	SERIAL_CONTROL_SERIAL6	SERIAL6
107	SERIAL_CONTROL_SERIAL7	SERIAL7
108	SERIAL_CONTROL_SERIAL8	SERIAL8
109	SERIAL_CONTROL_SERIAL9	SERIAL9

SERIAL_CONTROL_FLAG

[Enum] SERIAL_CONTROL flags (bitmask)

Value	Field Name	Description
1	SERIAL_CONTROL_FLAG_REPLY	Set if this is a reply
2	SERIAL_CONTROL_FLAG_RESPOND	Set if the sender wants the receiver to send a response as another SERIAL_CONTROL message
4	SERIAL_CONTROL_FLAG_EXCLUSIVE	Set if access to the serial port should be removed from whatever driver is currently using it, giving exclusive access to the SERIAL_CONTROL protocol. The port can be handed back by sending a request without this flag set
8	SERIAL_CONTROL_FLAG_BLOCKING	Block on writes to the serial port
16	SERIAL_CONTROL_FLAG_MULTI	Send multiple replies until port is drained

MAV_DISTANCE_SENSOR

[Enum] Enumeration of distance sensor types

Value	Field Name	Description
0	MAV_DISTANCE_SENSOR_LASER	Laser rangefinder, e.g. LightWare SF02/F or PulsedLight units
1	MAV_DISTANCE_SENSOR_ULTRASOUND	Ultrasound rangefinder, e.g. MaxBotix units
2	MAV_DISTANCE_SENSOR_INFRARED	Infrared rangefinder, e.g. Sharp units
3	MAV_DISTANCE_SENSOR_RADAR	Radar type, e.g. uLanding units
4	MAV_DISTANCE_SENSOR_UNKNOWN	Broken or unknown type, e.g. analog units

MAV_SENSOR_ORIENTATION

[Enum] Enumeration of sensor orientation, according to its rotations

Value	Field Name	Description
0	MAV_SENSOR_ROTATION_NONE	Roll: 0, Pitch: 0, Yaw: 0
1	MAV_SENSOR_ROTATION_YAW_45	Roll: 0, Pitch: 0, Yaw: 45
2	MAV_SENSOR_ROTATION_YAW_90	Roll: 0, Pitch: 0, Yaw: 90
3	MAV_SENSOR_ROTATION_YAW_135	Roll: 0, Pitch: 0, Yaw: 135
4	MAV_SENSOR_ROTATION_YAW_180	Roll: 0, Pitch: 0, Yaw: 180
5	MAV_SENSOR_ROTATION_YAW_225	Roll: 0, Pitch: 0, Yaw: 225
6	MAV_SENSOR_ROTATION_YAW_270	Roll: 0, Pitch: 0, Yaw: 270
7	MAV_SENSOR_ROTATION_YAW_315	Roll: 0, Pitch: 0, Yaw: 315
8	MAV_SENSOR_ROTATION_ROLL_180	Roll: 180, Pitch: 0, Yaw: 0
9	MAV_SENSOR_ROTATION_ROLL_180_YAW_45	Roll: 180, Pitch: 0, Yaw: 45
10	MAV_SENSOR_ROTATION_ROLL_180_YAW_90	Roll: 180, Pitch: 0, Yaw: 90
11	MAV_SENSOR_ROTATION_ROLL_180_YAW_135	Roll: 180, Pitch: 0, Yaw: 135
12	MAV_SENSOR_ROTATION_PITCH_180	Roll: 0, Pitch: 180, Yaw: 0
13	MAV_SENSOR_ROTATION_ROLL_180_YAW_225	Roll: 180, Pitch: 0, Yaw: 225
14	MAV_SENSOR_ROTATION_ROLL_180_YAW_270	Roll: 180, Pitch: 0, Yaw: 270
15	MAV_SENSOR_ROTATION_ROLL_180_YAW_315	Roll: 180, Pitch: 0, Yaw: 315
16	MAV_SENSOR_ROTATION_ROLL_90	Roll: 90, Pitch: 0, Yaw: 0
17	MAV_SENSOR_ROTATION_ROLL_90_YAW_45	Roll: 90, Pitch: 0, Yaw: 45
18	MAV_SENSOR_ROTATION_ROLL_90_YAW_90	Roll: 90, Pitch: 0, Yaw: 90
19	MAV_SENSOR_ROTATION_ROLL_90_YAW_135	Roll: 90, Pitch: 0, Yaw: 135
20	MAV_SENSOR_ROTATION_ROLL_270	Roll: 270, Pitch: 0, Yaw: 0
21	MAV_SENSOR_ROTATION_ROLL_270_YAW_45	Roll: 270, Pitch: 0, Yaw: 45
22	MAV_SENSOR_ROTATION_ROLL_270_YAW_90	Roll: 270, Pitch: 0, Yaw: 90
23	MAV_SENSOR_ROTATION_ROLL_270_YAW_135	Roll: 270, Pitch: 0, Yaw: 135
24	MAV_SENSOR_ROTATION_PITCH_90	Roll: 0, Pitch: 90, Yaw: 0
25	MAV_SENSOR_ROTATION_PITCH_270	Roll: 0, Pitch: 270, Yaw: 0
26	MAV_SENSOR_ROTATION_PITCH_180_YAW_90	Roll: 0, Pitch: 180, Yaw: 90
27	MAV_SENSOR_ROTATION_PITCH_180_YAW_270	Roll: 0, Pitch: 180, Yaw: 270
28	MAV_SENSOR_ROTATION_ROLL_90_PITCH_90	Roll: 90, Pitch: 90, Yaw: 0
29	MAV_SENSOR_ROTATION_ROLL_180_PITCH_90	Roll: 180, Pitch: 90, Yaw: 0
30	MAV_SENSOR_ROTATION_ROLL_270_PITCH_90	Roll: 270, Pitch: 90, Yaw: 0
31	MAV_SENSOR_ROTATION_ROLL_90_PITCH_180	Roll: 90, Pitch: 180, Yaw: 0
32	MAV_SENSOR_ROTATION_ROLL_270_PITCH_180	Roll: 270, Pitch: 180, Yaw: 0
33	MAV_SENSOR_ROTATION_ROLL_90_PITCH_270	Roll: 90, Pitch: 270, Yaw: 0
34	MAV_SENSOR_ROTATION_ROLL_180_PITCH_270	Roll: 180, Pitch: 270, Yaw: 0
35	MAV_SENSOR_ROTATION_ROLL_270_PITCH_270	Roll: 270, Pitch: 270, Yaw: 0
36	MAV_SENSOR_ROTATION_ROLL_90_PITCH_180_YAW_90	Roll: 90, Pitch: 180, Yaw: 90
37	MAV_SENSOR_ROTATION_ROLL_90_YAW_270	Roll: 90, Pitch: 0, Yaw: 270
38	MAV_SENSOR_ROTATION_ROLL_90_PITCH_68_YAW_293	Roll: 90, Pitch: 68, Yaw: 293
39	MAV_SENSOR_ROTATION_PITCH_315	Pitch: 315
40	MAV_SENSOR_ROTATION_ROLL_90_PITCH_315	Roll: 90, Pitch: 315
100	MAV_SENSOR_ROTATION_CUSTOM	Custom orientation

MAV_PROTOCOL_CAPABILITY

[Enum] Bitmask of (optional) autopilot capabilities (64 bit). If a bit is set, the autopilot supports this capability.

Value	Field Name	Description
1	MAV_PROTOCOL_CAPABILITY_MISSION_FLOAT	Autopilot supports the MISSION_ITEM float message type. Note that MISSION_ITEM is deprecated, and autopilots should use MISSION_INT instead.
2	MAV_PROTOCOL_CAPABILITY_PARAM_FLOAT	**DEPRECATED:** Replaced by MAV_PROTOCOL_CAPABILITY_PARAM_ENCODE_C_CAST (2022-03). | Autopilot supports the new param float message type. | | 4 | MAV_PROTOCOL_CAPABILITY_MISSION_INT | Autopilot supports MISSION_ITEM_INT scaled integer message type. Note that this flag must always be set if missions are supported, because missions must always use MISSION_ITEM_INT (rather than MISSION_ITEM, which is deprecated). | | 8 | MAV_PROTOCOL_CAPABILITY_COMMAND_INT | Autopilot supports COMMAND_INT scaled integer message type. | | 16 | MAV_PROTOCOL_CAPABILITY_PARAM_ENCODE_BYTEWISE | Parameter protocol uses byte-wise encoding of parameter values into param_value (float) fields: https://mavlink.io/en/services/parameter.html#parameter-encoding. Note that either this flag or MAV_PROTOCOL_CAPABILITY_PARAM_ENCODE_BYTEWISE should be set if the parameter protocol is supported. | | 32 | MAV_PROTOCOL_CAPABILITY_FTP | Autopilot supports the File Transfer Protocol v1: https://mavlink.io/en/services/ftp.html. | | 64 | MAV_PROTOCOL_CAPABILITY_SET_ATTITUDE_TARGET | Autopilot supports commanding attitude offboard. | | 128 | MAV_PROTOCOL_CAPABILITY_SET_POSITION_TARGET_LOCAL_NED | Autopilot supports commanding position and velocity targets in local NED frame. | | 256 | MAV_PROTOCOL_CAPABILITY_SET_POSITION_TARGET_GLOBAL_INT | Autopilot supports commanding position and velocity targets in global scaled integers. | | 512 | MAV_PROTOCOL_CAPABILITY_TERRAIN | Autopilot supports terrain protocol / data handling. | | 1024 | MAV_PROTOCOL_CAPABILITY_SET_ACTUATOR_TARGET | Autopilot supports direct actuator control. | | 2048 | MAV_PROTOCOL_CAPABILITY_FLIGHT_TERMINATION | Autopilot supports the MAV_CMD_DO_FLIGHTTERMINATION command (flight termination). | | 4096 | MAV_PROTOCOL_CAPABILITY_COMPASS_CALIBRATION | Autopilot supports onboard compass calibration. | | 8192 | MAV_PROTOCOL_CAPABILITY_MAVLINK2 | Autopilot supports MAVLink version 2. | | 16384 | MAV_PROTOCOL_CAPABILITY_MISSION_FENCE | Autopilot supports mission fence protocol. | | 32768 | MAV_PROTOCOL_CAPABILITY_MISSION_RALLY | Autopilot supports mission rally point protocol. | | 65536 | MAV_PROTOCOL_CAPABILITY_RESERVED2 | Reserved for future use. | | 131072 | MAV_PROTOCOL_CAPABILITY_PARAM_ENCODE_C_CAST | Parameter protocol uses C-cast of parameter values to set the param_value (float) fields: https://mavlink.io/en/services/parameter.html#parameter-encoding. Note that either this flag or MAV_PROTOCOL_CAPABILITY_PARAM_ENCODE_BYTEWISE should be set if the parameter protocol is supported. |

MAV_MISSION_TYPE

[Enum] Type of mission items being requested/sent in mission protocol.

Value	Field Name	Description

0	MAV_MISSION_TYPE_MISSION	Items are mission commands for main mission.
1	MAV_MISSION_TYPE_FENCE	Specifies GeoFence area(s). Items are MAV_CMD_NAV_FENCE_ GeoFence items.
2	MAV_MISSION_TYPE_RALLY	Specifies the rally points for the vehicle. Rally points are alternative RTL points. Items are MAV_CMD_NAV_RALLY_POINT rally point items.
255	MAV_MISSION_TYPE_ALL	Only used in MISSION_CLEAR_ALL to clear all mission types.

MAV_ESTIMATOR_TYPE

[Enum] Enumeration of estimator types

Value	Field Name	Description
0	MAV_ESTIMATOR_TYPE_UNKNOWN	Unknown type of the estimator.
1	MAV_ESTIMATOR_TYPE_NAIVE	This is a naive estimator without any real covariance feedback.
2	MAV_ESTIMATOR_TYPE_VISION	Computer vision based estimate. Might be up to scale.
3	MAV_ESTIMATOR_TYPE_VIO	Visual-inertial estimate.
4	MAV_ESTIMATOR_TYPE_GPS	Plain GPS estimate.
5	MAV_ESTIMATOR_TYPE_GPS_INS	Estimator integrating GPS and inertial sensing.
6	MAV_ESTIMATOR_TYPE_MOCAP	Estimate from external motion capturing system.
7	MAV_ESTIMATOR_TYPE_LIDAR	Estimator based on lidar sensor input.
8	MAV_ESTIMATOR_TYPE_AUTOPILOT	Estimator on autopilot.

MAV_BATTERY_TYPE

[Enum] Enumeration of battery types

Value	Field Name	Description
0	MAV_BATTERY_TYPE_UNKNOWN	Not specified.
1	MAV_BATTERY_TYPE_LIPO	Lithium polymer battery
2	MAV_BATTERY_TYPE_LIFE	Lithium-iron-phosphate battery
3	MAV_BATTERY_TYPE_LION	Lithium-ION battery
4	MAV_BATTERY_TYPE_NIMH	Nickel metal hydride battery

MAV_BATTERY_FUNCTION

[Enum] Enumeration of battery functions

Value	Field Name	Description
0	MAV_BATTERY_FUNCTION_UNKNOWN	Battery function is unknown
1	MAV_BATTERY_FUNCTION_ALL	Battery supports all flight systems
2	MAV_BATTERY_FUNCTION_PROPULSION	Battery for the propulsion system
3	MAV_BATTERY_FUNCTION_AVIONICS	Avionics battery
4	MAV_BATTERY_TYPE_PAYLOAD	Payload battery

MAV_BATTERY_CHARGE_STATE

[Enum] Enumeration for battery charge states.

Value	Field Name	Description
0	MAV_BATTERY_CHARGE_STATE_UNDEFINED	Low battery state is not provided
1	MAV_BATTERY_CHARGE_STATE_OK	Battery is not in low state. Normal operation.
2	MAV_BATTERY_CHARGE_STATE_LOW	Battery state is low, warn and monitor close.
3	MAV_BATTERY_CHARGE_STATE_CRITICAL	Battery state is critical, return or abort immediately.
4	MAV_BATTERY_CHARGE_STATE_EMERGENCY	Battery state is too low for ordinary abort sequence. Perform fastest possible emergency stop to prevent damage.
5	MAV_BATTERY_CHARGE_STATE_FAILED	Battery failed, damage unavoidable. Possible causes (faults) are listed in MAV_BATTERY_FAULT.
6	MAV_BATTERY_CHARGE_STATE_UNHEALTHY	Battery is diagnosed to be defective or an error occurred, usage is discouraged / prohibited. Possible causes (faults) are listed in MAV_BATTERY_FAULT.
7	MAV_BATTERY_CHARGE_STATE_CHARGING	Battery is charging.

MAV_BATTERY_MODE

[Enum] Battery mode. Note, the normal operation mode (i.e. when flying) should be reported as MAV_BATTERY_MODE_UNKNOWN to allow message trimming in normal flight.

Value	Field Name	Description
0	MAV_BATTERY_MODE_UNKNOWN	Battery mode not supported/unknown battery mode/normal operation.
1	MAV_BATTERY_MODE_AUTO_DISCHARGING	Battery is auto discharging (towards storage level).
2	MAV_BATTERY_MODE_HOT_SWAP	Battery in hot-swap mode (current limited to prevent spikes that might damage sensitive electrical circuits).

MAV_BATTERY_FAULT

[Enum] Smart battery supply status/fault flags (bitmask) for health indication. The battery must also report either MAV_BATTERY_CHARGE_STATE_FAILED or MAV_BATTERY_CHARGE_STATE_UNHEALTHY if any of these are set.

Value	Field Name	Description
1	MAV_BATTERY_FAULT_DEEP_DISCHARGE	Battery has deep discharged.
2	MAV_BATTERY_FAULT_SPIKES	Voltage spikes.
4	MAV_BATTERY_FAULT_CELL_FAIL	One or more cells have failed. Battery should also report MAV_BATTERY_CHARGE_STATE_FAILE (and should not be used).
8	MAV_BATTERY_FAULT_OVER_CURRENT	Over-current fault.
16	MAV_BATTERY_FAULT_OVER_TEMPERATURE	Over-temperature fault.
32	MAV_BATTERY_FAULT_UNDER_TEMPERATURE	Under-temperature fault.

64	MAV_BATTERY_FAULT_INCOMPATIBLE_VOLTAGE	Vehicle voltage is not compatible with this battery (batteries on same power rail should have similar voltage).
128	MAV_BATTERY_FAULT_INCOMPATIBLE_FIRMWARE	Battery firmware is not compatible with current autopilot firmware.
256	BATTERY_FAULT_INCOMPATIBLE_CELLS_CONFIGURATION	Battery is not compatible due to cell configuration (e.g. 5s1p when vehicle requires 6s).

MAV_GENERATOR_STATUS_FLAG

[Enum] Flags to report status/failure cases for a power generator (used in GENERATOR_STATUS). Note that FAULTS are conditions that cause the generator to fail. Warnings are conditions that require attention before the next use (they indicate the system is not operating properly).

Value	Field Name	Description
1	MAV_GENERATOR_STATUS_FLAG_OFF	Generator is off.
2	MAV_GENERATOR_STATUS_FLAG_READY	Generator is ready to start generating power.
4	MAV_GENERATOR_STATUS_FLAG_GENERATING	Generator is generating power.
8	MAV_GENERATOR_STATUS_FLAG_CHARGING	Generator is charging the batteries (generating enough power to charge and provide the load).
16	MAV_GENERATOR_STATUS_FLAG_REDUCED_POWER	Generator is operating at a reduced maximum power.
32	MAV_GENERATOR_STATUS_FLAG_MAXPOWER	Generator is providing the maximum output.
64	MAV_GENERATOR_STATUS_FLAG_OVERTEMP_WARNING	Generator is near the maximum operating temperature, cooling is insufficient.
128	MAV_GENERATOR_STATUS_FLAG_OVERTEMP_FAULT	Generator hit the maximum operating temperature and shutdown.
256	MAV_GENERATOR_STATUS_FLAG_ELECTRONICS_OVERTEMP_WARNING	Power electronics are near the maximum operating temperature, cooling is insufficient.
512	MAV_GENERATOR_STATUS_FLAG_ELECTRONICS_OVERTEMP_FAULT	Power electronics hit the maximum operating temperature and shutdown.
1024	MAV_GENERATOR_STATUS_FLAG_ELECTRONICS_FAULT	Power electronics experienced a fault and shutdown.
2048	MAV_GENERATOR_STATUS_FLAG_POWERSOURCE_FAULT	The power source supplying the generator failed e.g. mechanical generator stopped, tether is no longer providing power, solar cell is in shade, hydrogen reaction no longer happening.
4096	MAV_GENERATOR_STATUS_FLAG_COMMUNICATION_WARNING	Generator controller having communication problems.
8192	MAV_GENERATOR_STATUS_FLAG_COOLING_WARNING	Power electronic or generator cooling system error.
16384	MAV_GENERATOR_STATUS_FLAG_POWER_RAIL_FAULT	Generator controller power rail experienced a fault.
32768	MAV_GENERATOR_STATUS_FLAG_OVERCURRENT_FAULT	Generator controller exceeded the overcurrent threshold and shutdown to prevent damage.
65536	MAV_GENERATOR_STATUS_FLAG_BATTERY_OVERCHARGE_CURRENT_FAULT	Generator controller detected a high current going into the batteries and shutdown to prevent battery damage.
131072	MAV_GENERATOR_STATUS_FLAG_OVERVOLTAGE_FAULT	Generator controller exceeded it's overvoltage threshold and shutdown to prevent it exceeding the voltage rating.
262144	MAV_GENERATOR_STATUS_FLAG_BATTERY_UNDERVOLT_FAULT	Batteries are under voltage (generator will not start).
524288	MAV_GENERATOR_STATUS_FLAG_START_INHIBITED	Generator start is inhibited by e.g. a safety switch.
1048576	MAV_GENERATOR_STATUS_FLAG_MAINTENANCE_REQUIRED	Generator requires maintenance.
2097152	MAV_GENERATOR_STATUS_FLAG_WARMING_UP	Generator is not ready to generate yet.
4194304	MAV_GENERATOR_STATUS_FLAG_IDLE	Generator is idle.

MAV_VTOL_STATE

[Enum] Enumeration of VTOL states

Value	Field Name	Description
0	MAV_VTOL_STATE_UNDEFINED	MAV is not configured as VTOL
1	MAV_VTOL_STATE_TRANSITION_TO_FW	VTOL is in transition from multicopter to fixed-wing
2	MAV_VTOL_STATE_TRANSITION_TO_MC	VTOL is in transition from fixed-wing to multicopter
3	MAV_VTOL_STATE_MC	VTOL is in multicopter state
4	MAV_VTOL_STATE_FW	VTOL is in fixed-wing state

MAV_LANDED_STATE

[Enum] Enumeration of landed detector states

Value	Field Name	Description
0	MAV_LANDED_STATE_UNDEFINED	MAV landed state is unknown
1	MAV_LANDED_STATE_ON_GROUND	MAV is landed (on ground)
2	MAV_LANDED_STATE_IN_AIR	MAV is in air
3	MAV_LANDED_STATE_TAKEOFF	MAV currently taking off
4	MAV_LANDED_STATE_LANDING	MAV currently landing

ADSB_ALTITUDE_TYPE

[Enum] Enumeration of the ADSB altimeter types

Value	Field Name	Description
0	ADSB_ALTITUDE_TYPE_PRESSURE_QNH	Altitude reported from a Baro source using QNH reference
1	ADSB_ALTITUDE_TYPE_GEOMETRIC	Altitude reported from a GNSS source

ADSB_EMITTER_TYPE

[Enum] ADSB classification for the type of vehicle emitting the transponder signal

Value	Field Name	Description
0	ADSB_EMITTER_TYPE_NO_INFO	
1	ADSB_EMITTER_TYPE_LIGHT	
2	ADSB_EMITTER_TYPE_SMALL	
3	ADSB_EMITTER_TYPE_LARGE	
4	ADSB_EMITTER_TYPE_HIGH_VORTEX_LARGE	
5	ADSB_EMITTER_TYPE_HEAVY	
6	ADSB_EMITTER_TYPE_HIGHLY_MANUV	
7	ADSB_EMITTER_TYPE_ROTOCRAFT	
8	ADSB_EMITTER_TYPE_UNASSIGNED	
9	ADSB_EMITTER_TYPE_GLIDER	
10	ADSB_EMITTER_TYPE_LIGHTER_AIR	
11	ADSB_EMITTER_TYPE_PARACHUTE	
12	ADSB_EMITTER_TYPE_ULTRA_LIGHT	
13	ADSB_EMITTER_TYPE_UNASSIGNED2	
14	ADSB_EMITTER_TYPE_UAV	
15	ADSB_EMITTER_TYPE_SPACE	
16	ADSB_EMITTER_TYPE_UNASSGINED3	
17	ADSB_EMITTER_TYPE_EMERGENCY_SURFACE	
18	ADSB_EMITTER_TYPE_SERVICE_SURFACE	
19	ADSB_EMITTER_TYPE_POINT_OBSTACLE	

ADSB_FLAGS

[Enum] These flags indicate status such as data validity of each data source. Set = data valid

Value	Field Name	Description
1	ADSB_FLAGS_VALID_COORDS	
2	ADSB_FLAGS_VALID_ALTITUDE	
4	ADSB_FLAGS_VALID_HEADING	
8	ADSB_FLAGS_VALID_VELOCITY	
16	ADSB_FLAGS_VALID_CALLSIGN	
32	ADSB_FLAGS_VALID_SQUAWK	
64	ADSB_FLAGS_SIMULATED	
128	ADSB_FLAGS_VERTICAL_VELOCITY_VALID	
256	ADSB_FLAGS_BARO_VALID	
32768	ADSB_FLAGS_SOURCE_UAT	

MAV_DO_REPOSITION_FLAGS

[Enum] Bitmap of options for the MAV_CMD_DO_REPOSITION

Value	Field Name	Description
1	MAV_DO_REPOSITION_FLAGS_CHANGE_MODE	The aircraft should immediately transition into guided. This should not be set for follow me applications

ESTIMATOR_STATUS_FLAGS

[Enum] Flags in ESTIMATOR_STATUS message

Value	Field Name	Description
1	ESTIMATOR_ATTITUDE	True if the attitude estimate is good
2	ESTIMATOR_VELOCITY_HORIZ	True if the horizontal velocity estimate is good
4	ESTIMATOR_VELOCITY_VERT	True if the vertical velocity estimate is good
8	ESTIMATOR_POS_HORIZ_REL	True if the horizontal position (relative) estimate is good
16	ESTIMATOR_POS_HORIZ_ABS	True if the horizontal position (absolute) estimate is good
32	ESTIMATOR_POS_VERT_ABS	True if the vertical position (absolute) estimate is good
64	ESTIMATOR_POS_VERT_AGL	True if the vertical position (above ground) estimate is good
128	ESTIMATOR_CONST_POS_MODE	True if the EKF is in a constant position mode and is not using external measurements (eg GPS or optical flow)
256	ESTIMATOR_PRED_POS_HORIZ_REL	True if the EKF has sufficient data to enter a mode that will provide a (relative) position estimate
512	ESTIMATOR_PRED_POS_HORIZ_ABS	True if the EKF has sufficient data to enter a mode that will provide a (absolute) position estimate
1024	ESTIMATOR_GPS_GLITCH	True if the EKF has detected a GPS glitch
2048	ESTIMATOR_ACCEL_ERROR	True if the EKF has detected bad accelerometer data

MOTOR_TEST_ORDER

[Enum] Sequence that motors are tested when using MAV_CMD_DO_MOTOR_TEST.

Value	Field Name	Description
0	MOTOR_TEST_ORDER_DEFAULT	Default autopilot motor test method.
1	MOTOR_TEST_ORDER_SEQUENCE	Motor numbers are specified as their index in a predefined vehicle-specific sequence.

2	MOTOR_TEST_ORDER_BOARD	Motor numbers are specified as the output as labeled on the board.

MOTOR_TEST_THROTTLE_TYPE

[Enum] Defines how throttle value is represented in MAV_CMD_DO_MOTOR_TEST.

Value	Field Name	Description
0	MOTOR_TEST_THROTTLE_PERCENT	Throttle as a percentage (0 ~ 100)
1	MOTOR_TEST_THROTTLE_PWM	Throttle as an absolute PWM value (normally in range of 1000~2000).
2	MOTOR_TEST_THROTTLE_PILOT	Throttle pass-through from pilot's transmitter.
3	MOTOR_TEST_COMPASS_CAL	Per-motor compass calibration test.

GPS_INPUT_IGNORE_FLAGS

[Enum]

Value	Field Name	Description
1	GPS_INPUT_IGNORE_FLAG_ALT	ignore altitude field
2	GPS_INPUT_IGNORE_FLAG_HDOP	ignore hdop field
4	GPS_INPUT_IGNORE_FLAG_VDOP	ignore vdop field
8	GPS_INPUT_IGNORE_FLAG_VEL_HORIZ	ignore horizontal velocity field (vn and ve)
16	GPS_INPUT_IGNORE_FLAG_VEL_VERT	ignore vertical velocity field (vd)
32	GPS_INPUT_IGNORE_FLAG_SPEED_ACCURACY	ignore speed accuracy field
64	GPS_INPUT_IGNORE_FLAG_HORIZONTAL_ACCURACY	ignore horizontal accuracy field
128	GPS_INPUT_IGNORE_FLAG_VERTICAL_ACCURACY	ignore vertical accuracy field

MAV_COLLISION_ACTION

[Enum] Possible actions an aircraft can take to avoid a collision.

Value	Field Name	Description
0	MAV_COLLISION_ACTION_NONE	Ignore any potential collisions
1	MAV_COLLISION_ACTION_REPORT	Report potential collision
2	MAV_COLLISION_ACTION_ASCEND_OR_DESCEND	Ascend or Descend to avoid threat
3	MAV_COLLISION_ACTION_MOVE_HORIZONTALLY	Move horizontally to avoid threat
4	MAV_COLLISION_ACTION_MOVE_PERPENDICULAR	Aircraft to move perpendicular to the collision's velocity vector
5	MAV_COLLISION_ACTION_RTL	Aircraft to fly directly back to its launch point
6	MAV_COLLISION_ACTION_HOVER	Aircraft to stop in place

MAV_COLLISION_THREAT_LEVEL

[Enum] Aircraft-rated danger from this threat.

Value	Field Name	Description
0	MAV_COLLISION_THREAT_LEVEL_NONE	Not a threat
1	MAV_COLLISION_THREAT_LEVEL_LOW	Craft is mildly concerned about this threat
2	MAV_COLLISION_THREAT_LEVEL_HIGH	Craft is panicking, and may take actions to avoid threat

MAV_COLLISION_SRC

[Enum] Source of information about this collision.

Value	Field Name	Description
0	MAV_COLLISION_SRC_ADSB	ID field references ADSB_VEHICLE packets
1	MAV_COLLISION_SRC_MAVLINK_GPS_GLOBAL_INT	ID field references MAVLink SRC ID

GPS_FIX_TYPE

[Enum] Type of GPS fix

Value	Field Name	Description
0	GPS_FIX_TYPE_NO_GPS	No GPS connected
1	GPS_FIX_TYPE_NO_FIX	No position information, GPS is connected
2	GPS_FIX_TYPE_2D_FIX	2D position
3	GPS_FIX_TYPE_3D_FIX	3D position
4	GPS_FIX_TYPE_DGPS	DGPS/SBAS aided 3D position
5	GPS_FIX_TYPE_RTK_FLOAT	RTK float, 3D position
6	GPS_FIX_TYPE_RTK_FIXED	RTK Fixed, 3D position
7	GPS_FIX_TYPE_STATIC	Static fixed, typically used for base stations
8	GPS_FIX_TYPE_PPP	PPP, 3D position.

RTK_BASELINE_COORDINATE_SYSTEM

[Enum] RTK GPS baseline coordinate system, used for RTK corrections

Value	Field Name	Description
0	RTK_BASELINE_COORDINATE_SYSTEM_ECEF	Earth-centered, Earth-fixed
1	RTK_BASELINE_COORDINATE_SYSTEM_NED	RTK basestation centered, north, east, down

LANDING_TARGET_TYPE

[Enum] Type of landing target

Value	Field Name	Description
0	LANDING_TARGET_TYPE_LIGHT_BEACON	

		Landing target signaled by light beacon (ex: IR-LOCK)
1	LANDING_TARGET_TYPE_RADIO_BEACON	Landing target signaled by radio beacon (ex: ILS, NDB)
2	LANDING_TARGET_TYPE_VISION_FIDUCIAL	Landing target represented by a fiducial marker (ex: ARTag)
3	LANDING_TARGET_TYPE_VISION_OTHER	Landing target represented by a pre-defined visual shape/feature (ex: X-marker, H-marker, square)

VTOL_TRANSITION_HEADING

[Enum] Direction of VTOL transition

Value	Field Name	Description
0	VTOL_TRANSITION_HEADING_VEHICLE_DEFAULT	Respect the heading configuration of the vehicle.
1	VTOL_TRANSITION_HEADING_NEXT_WAYPOINT	Use the heading pointing towards the next waypoint.
2	VTOL_TRANSITION_HEADING_TAKEOFF	Use the heading on takeoff (while sitting on the ground).
3	VTOL_TRANSITION_HEADING_SPECIFIED	Use the specified heading in parameter 4.
4	VTOL_TRANSITION_HEADING_ANY	Use the current heading when reaching takeoff altitude (potentially facing the wind when weather-vaning is active).

CAMERA_CAP_FLAGS

[Enum] Camera capability flags (Bitmap)

Value	Field Name	Description
1	CAMERA_CAP_FLAGS_CAPTURE_VIDEO	Camera is able to record video
2	CAMERA_CAP_FLAGS_CAPTURE_IMAGE	Camera is able to capture images
4	CAMERA_CAP_FLAGS_HAS_MODES	Camera has separate Video and Image/Photo modes (MAV_CMD_SET_CAMERA_MODE)
8	CAMERA_CAP_FLAGS_CAN_CAPTURE_IMAGE_IN_VIDEO_MODE	Camera can capture images while in video mode
16	CAMERA_CAP_FLAGS_CAN_CAPTURE_VIDEO_IN_IMAGE_MODE	Camera can capture videos while in Photo/Image mode
32	CAMERA_CAP_FLAGS_HAS_IMAGE_SURVEY_MODE	Camera has image survey mode (MAV_CMD_SET_CAMERA_MODE)
64	CAMERA_CAP_FLAGS_HAS_BASIC_ZOOM	Camera has basic zoom control (MAV_CMD_SET_CAMERA_ZOOM)
128	CAMERA_CAP_FLAGS_HAS_BASIC_FOCUS	Camera has basic focus control (MAV_CMD_SET_CAMERA_FOCUS)
256	CAMERA_CAP_FLAGS_HAS_VIDEO_STREAM	Camera has video streaming capabilities (request VIDEO_STREAM_INFORMATION with MAV_CMD_REQUEST_MESSAGE for video streaming info)
512	CAMERA_CAP_FLAGS_HAS_TRACKING_POINT	Camera supports tracking of a point on the camera view.
1024	CAMERA_CAP_FLAGS_HAS_TRACKING_RECTANGLE	Camera supports tracking of a selection rectangle on the camera view.
2048	CAMERA_CAP_FLAGS_HAS_TRACKING_GEO_STATUS	Camera supports tracking geo status (CAMERA_TRACKING_GEO_STATUS).

VIDEO_STREAM_STATUS_FLAGS

[Enum] Stream status flags (Bitmap)

Value	Field Name	Description
1	VIDEO_STREAM_STATUS_FLAGS_RUNNING	Stream is active (running)
2	VIDEO_STREAM_STATUS_FLAGS_THERMAL	Stream is thermal imaging

VIDEO_STREAM_TYPE

[Enum] Video stream types

Value	Field Name	Description
0	VIDEO_STREAM_TYPE_RTSP	Stream is RTSP
1	VIDEO_STREAM_TYPE_RTPUDP	Stream is RTP UDP (URI gives the port number)
2	VIDEO_STREAM_TYPE_TCP_MPEG	Stream is MPEG on TCP
3	VIDEO_STREAM_TYPE_MPEG_TS_H264	Stream is h.264 on MPEG TS (URI gives the port number)

CAMERA_TRACKING_STATUS_FLAGS

[Enum] Camera tracking status flags

Value	Field Name	Description
0	CAMERA_TRACKING_STATUS_FLAGS_IDLE	Camera is not tracking
1	CAMERA_TRACKING_STATUS_FLAGS_ACTIVE	Camera is tracking
2	CAMERA_TRACKING_STATUS_FLAGS_ERROR	Camera tracking in error state

CAMERA_TRACKING_MODE

[Enum] Camera tracking modes

Value	Field Name	Description
0	CAMERA_TRACKING_MODE_NONE	Not tracking
1	CAMERA_TRACKING_MODE_POINT	Target is a point
2	CAMERA_TRACKING_MODE_RECTANGLE	Target is a rectangle

CAMERA_TRACKING_TARGET_DATA

[Enum] Camera tracking target data (shows where tracked target is within image)

Value	Field Name	Description

0	CAMERA_TRACKING_TARGET_DATA_NONE	No target data
1	CAMERA_TRACKING_TARGET_DATA_EMBEDDED	Target data embedded in image data (proprietary)
2	CAMERA_TRACKING_TARGET_DATA_RENDERED	Target data rendered in image
4	CAMERA_TRACKING_TARGET_DATA_IN_STATUS	Target data within status message (Point or Rectangle)

CAMERA_ZOOM_TYPE

[Enum] Zoom types for MAV_CMD_SET_CAMERA_ZOOM

Value	Field Name	Description
0	ZOOM_TYPE_STEP	Zoom one step increment (-1 for wide, 1 for tele)
1	ZOOM_TYPE_CONTINUOUS	Continuous zoom up/down until stopped (-1 for wide, 1 for tele, 0 to stop zooming)
2	ZOOM_TYPE_RANGE	Zoom value as proportion of full camera range (a value between 0.0 and 100.0)
3	ZOOM_TYPE_FOCAL_LENGTH	Zoom value/variable focal length in millimetres. Note that there is no message to get the valid zoom range of the camera, so this can type can only be used for cameras where the zoom range is known (implying that this cannot reliably be used in a GCS for an arbitrary camera)

SET_FOCUS_TYPE

[Enum] Focus types for MAV_CMD_SET_CAMERA_FOCUS

Value	Field Name	Description
0	FOCUS_TYPE_STEP	Focus one step increment (-1 for focusing in, 1 for focusing out towards infinity).
1	FOCUS_TYPE_CONTINUOUS	Continuous focus up/down until stopped (-1 for focusing in, 1 for focusing out towards infinity, 0 to stop focusing)
2	FOCUS_TYPE_RANGE	Focus value as proportion of full camera focus range (a value between 0.0 and 100.0)
3	FOCUS_TYPE_METERS	Focus value in metres. Note that there is no message to get the valid focus range of the camera, so this can type can only be used for cameras where the range is known (implying that this cannot reliably be used in a GCS for an arbitrary camera).
4	FOCUS_TYPE_AUTO	Focus automatically.
5	FOCUS_TYPE_AUTO_SINGLE	Single auto focus. Mainly used for still pictures. Usually abbreviated as AF-S.
6	FOCUS_TYPE_AUTO_CONTINUOUS	Continuous auto focus. Mainly used for dynamic scenes. Abbreviated as AF-C.

PARAM_ACK

[Enum] Result from PARAM_EXT_SET message (or a PARAM_SET within a transaction).

Value	Field Name	Description
0	PARAM_ACK_ACCEPTED	Parameter value ACCEPTED and SET
1	PARAM_ACK_VALUE_UNSUPPORTED	Parameter value UNKNOWN/UNSUPPORTED
2	PARAM_ACK_FAILED	Parameter failed to set
3	PARAM_ACK_IN_PROGRESS	Parameter value received but not yet set/accepted. A subsequent PARAM_ACK_TRANSACTION or PARAM_EXT_ACK with the final result will follow once operation is completed. This is returned immediately for parameters that take longer to set, indicating that the the parameter was received and does not need to be resent.

CAMERA_MODE

[Enum] Camera Modes.

Value	Field Name	Description
0	CAMERA_MODE_IMAGE	Camera is in image/photo capture mode.
1	CAMERA_MODE_VIDEO	Camera is in video capture mode.
2	CAMERA_MODE_IMAGE_SURVEY	Camera is in image survey capture mode. It allows for camera controller to do specific settings for surveys.

MAV_ARM_AUTH_DENIED_REASON

[Enum]

Value	Field Name	Description
0	MAV_ARM_AUTH_DENIED_REASON_GENERIC	Not a specific reason
1	MAV_ARM_AUTH_DENIED_REASON_NONE	Authorizer will send the error as string to GCS
2	MAV_ARM_AUTH_DENIED_REASON_INVALID_WAYPOINT	At least one waypoint have a invalid value
3	MAV_ARM_AUTH_DENIED_REASON_TIMEOUT	Timeout in the authorizer process(in case it depends on network)
4	MAV_ARM_AUTH_DENIED_REASON_AIRSPACE_IN_USE	Airspace of the mission in use by another vehicle, second result parameter can have the waypoint id that caused it to be denied.
5	MAV_ARM_AUTH_DENIED_REASON_BAD_WEATHER	Weather is not good to fly

RC_TYPE

[Enum] RC type

Value	Field Name	Description
0	RC_TYPE_SPEKTRUM_DSM2	Spektrum DSM2
1	RC_TYPE_SPEKTRUM_DSMX	Spektrum DSMX

POSITION_TARGET_TYPEMASK

[Enum] Bitmap to indicate which dimensions should be ignored by the vehicle: a value of 0b0000000000000000 or 0b0000001000000000 indicates that none of the setpoint dimensions should be ignored. If bit 9 is set the floats afx afy afz should be interpreted as force instead of acceleration.

Value	Field Name	Description
1	POSITION_TARGET_TYPEMASK_X_IGNORE	Ignore position x
2	POSITION_TARGET_TYPEMASK_Y_IGNORE	Ignore position y

4	POSITION_TARGET_TYPEMASK_Z_IGNORE	Ignore position z
8	POSITION_TARGET_TYPEMASK_VX_IGNORE	Ignore velocity x
16	POSITION_TARGET_TYPEMASK_VY_IGNORE	Ignore velocity y
32	POSITION_TARGET_TYPEMASK_VZ_IGNORE	Ignore velocity z
64	POSITION_TARGET_TYPEMASK_AX_IGNORE	Ignore acceleration x
128	POSITION_TARGET_TYPEMASK_AY_IGNORE	Ignore acceleration y
256	POSITION_TARGET_TYPEMASK_AZ_IGNORE	Ignore acceleration z
512	POSITION_TARGET_TYPEMASK_FORCE_SET	Use force instead of acceleration
1024	POSITION_TARGET_TYPEMASK_YAW_IGNORE	Ignore yaw
2048	POSITION_TARGET_TYPEMASK_YAW_RATE_IGNORE	Ignore yaw rate

ATTITUDE_TARGET_TYPEMASK

[Enum] Bitmap to indicate which dimensions should be ignored by the vehicle: a value of 0b00000000 indicates that none of the setpoint dimensions should be ignored.

Value	Field Name	Description
1	ATTITUDE_TARGET_TYPEMASK_BODY_ROLL_RATE_IGNORE	Ignore body roll rate
2	ATTITUDE_TARGET_TYPEMASK_BODY_PITCH_RATE_IGNORE	Ignore body pitch rate
4	ATTITUDE_TARGET_TYPEMASK_BODY_YAW_RATE_IGNORE	Ignore body yaw rate
32	ATTITUDE_TARGET_TYPEMASK_THRUST_BODY_SET	Use 3D body thrust setpoint instead of throttle
64	ATTITUDE_TARGET_TYPEMASK_THROTTLE_IGNORE	Ignore throttle
128	ATTITUDE_TARGET_TYPEMASK_ATTITUDE_IGNORE	Ignore attitude

UTM_FLIGHT_STATE

[Enum] Airborne status of UAS.

Value	Field Name	Description
1	UTM_FLIGHT_STATE_UNKNOWN	The flight state can't be determined.
2	UTM_FLIGHT_STATE_GROUND	UAS on ground.
3	UTM_FLIGHT_STATE_AIRBORNE	UAS airborne.
16	UTM_FLIGHT_STATE_EMERGENCY	UAS is in an emergency flight state.
32	UTM_FLIGHT_STATE_NOCTRL	UAS has no active controls.

UTM_DATA_AVAIL_FLAGS

[Enum] Flags for the global position report.

Value	Field Name	Description
1	UTM_DATA_AVAIL_FLAGS_TIME_VALID	The field time contains valid data.
2	UTM_DATA_AVAIL_FLAGS_UAS_ID_AVAILABLE	The field uas_id contains valid data.
4	UTM_DATA_AVAIL_FLAGS_POSITION_AVAILABLE	The fields lat, lon and h_acc contain valid data.
8	UTM_DATA_AVAIL_FLAGS_ALTITUDE_AVAILABLE	The fields alt and v_acc contain valid data.
16	UTM_DATA_AVAIL_FLAGS_RELATIVE_ALTITUDE_AVAILABLE	The field relative_alt contains valid data.
32	UTM_DATA_AVAIL_FLAGS_HORIZONTAL_VELO_AVAILABLE	The fields vx and vy contain valid data.
64	UTM_DATA_AVAIL_FLAGS_VERTICAL_VELO_AVAILABLE	The field vz contains valid data.
128	UTM_DATA_AVAIL_FLAGS_NEXT_WAYPOINT_AVAILABLE	The fields next_lat, next_lon and next_alt contain valid data.

CELLULAR_STATUS_FLAG

[Enum] These flags encode the cellular network status

Value	Field Name	Description
0	CELLULAR_STATUS_FLAG_UNKNOWN	State unknown or not reportable.
1	CELLULAR_STATUS_FLAG_FAILED	Modem is unusable
2	CELLULAR_STATUS_FLAG_INITIALIZING	Modem is being initialized
3	CELLULAR_STATUS_FLAG_LOCKED	Modem is locked
4	CELLULAR_STATUS_FLAG_DISABLED	Modem is not enabled and is powered down
5	CELLULAR_STATUS_FLAG_DISABLING	Modem is currently transitioning to the CELLULAR_STATUS_FLAG_DISABLED state
6	CELLULAR_STATUS_FLAG_ENABLING	Modem is currently transitioning to the CELLULAR_STATUS_FLAG_ENABLED state
7	CELLULAR_STATUS_FLAG_ENABLED	Modem is enabled and powered on but not registered with a network provider and not available for data connections
8	CELLULAR_STATUS_FLAG_SEARCHING	Modem is searching for a network provider to register
9	CELLULAR_STATUS_FLAG_REGISTERED	Modem is registered with a network provider, and data connections and messaging may be available for use
10	CELLULAR_STATUS_FLAG_DISCONNECTING	Modem is disconnecting and deactivating the last active packet data bearer. This state will not be entered if more than one packet data bearer is active and one of the active bearers is deactivated
11	CELLULAR_STATUS_FLAG_CONNECTING	Modem is activating and connecting the first packet data bearer. Subsequent bearer activations when another bearer is already active do not cause this state to be entered
12	CELLULAR_STATUS_FLAG_CONNECTED	One or more packet data bearers is active and connected

CELLULAR_NETWORK_FAILED_REASON

[Enum] These flags are used to diagnose the failure state of CELLULAR_STATUS

Value	Field Name	Description
0	CELLULAR_NETWORK_FAILED_REASON_NONE	No error
1	CELLULAR_NETWORK_FAILED_REASON_UNKNOWN	Error state is unknown
2	CELLULAR_NETWORK_FAILED_REASON_SIM_MISSING	SIM is required for the modem but missing

3	CELLULAR_NETWORK_FAILED_REASON_SIM_ERROR	SIM is available, but not usable for connection

CELLULAR_NETWORK_RADIO_TYPE

[Enum] Cellular network radio type

Value	Field Name	Description
0	CELLULAR_NETWORK_RADIO_TYPE_NONE	
1	CELLULAR_NETWORK_RADIO_TYPE_GSM	
2	CELLULAR_NETWORK_RADIO_TYPE_CDMA	
3	CELLULAR_NETWORK_RADIO_TYPE_WCDMA	
4	CELLULAR_NETWORK_RADIO_TYPE_LTE	

PRECISION_LAND_MODE

[Enum] Precision land modes (used in MAV_CMD_NAV_LAND).

Value	Field Name	Description
0	PRECISION_LAND_MODE_DISABLED	Normal (non-precision) landing.
1	PRECISION_LAND_MODE_OPPORTUNISTIC	Use precision landing if beacon detected when land command accepted, otherwise land normally.
2	PRECISION_LAND_MODE_REQUIRED	Use precision landing, searching for beacon if not found when land command accepted (land normally if beacon cannot be found).

PARACHUTE_ACTION

[Enum] Parachute actions. Trigger release and enable/disable auto-release.

Value	Field Name	Description
0	PARACHUTE_DISABLE	Disable auto-release of parachute (i.e. release triggered by crash detectors).
1	PARACHUTE_ENABLE	Enable auto-release of parachute.
2	PARACHUTE_RELEASE	Release parachute and kill motors.

MAV_TUNNEL_PAYLOAD_TYPE

[Enum]

Value	Field Name	Description
0	MAV_TUNNEL_PAYLOAD_TYPE_UNKNOWN	Encoding of payload unknown.
200	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED0	Registered for STorM32 gimbal controller.
201	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED1	Registered for STorM32 gimbal controller.
202	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED2	Registered for STorM32 gimbal controller.
203	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED3	Registered for STorM32 gimbal controller.
204	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED4	Registered for STorM32 gimbal controller.
205	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED5	Registered for STorM32 gimbal controller.
206	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED6	Registered for STorM32 gimbal controller.
207	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED7	Registered for STorM32 gimbal controller.
208	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED8	Registered for STorM32 gimbal controller.
209	MAV_TUNNEL_PAYLOAD_TYPE_STORM32_RESERVED9	Registered for STorM32 gimbal controller.

MAV_ODID_ID_TYPE

[Enum]

Value	Field Name	Description
0	MAV_ODID_ID_TYPE_NONE	No type defined.
1	MAV_ODID_ID_TYPE_SERIAL_NUMBER	Manufacturer Serial Number (ANSI/CTA-2063 format).
2	MAV_ODID_ID_TYPE_CAA_REGISTRATION_ID	CAA (Civil Aviation Authority) registered ID. Format: [ICAO Country Code].[CAA Assigned ID].
3	MAV_ODID_ID_TYPE_UTM_ASSIGNED_UUID	UTM (Unmanned Traffic Management) assigned UUID (RFC4122).
4	MAV_ODID_ID_TYPE_SPECIFIC_SESSION_ID	A 20 byte ID for a specific flight/session. The exact ID type is indicated by the first byte of uas_id and these type values are managed by ICAO.

MAV_ODID_UA_TYPE

[Enum]

Value	Field Name	Description
0	MAV_ODID_UA_TYPE_NONE	No UA (Unmanned Aircraft) type defined.
1	MAV_ODID_UA_TYPE_AEROPLANE	Aeroplane/Airplane. Fixed wing.
2	MAV_ODID_UA_TYPE_HELICOPTER_OR_MULTIROTOR	Helicopter or multirotor.
3	MAV_ODID_UA_TYPE_GYROPLANE	Gyroplane.
4	MAV_ODID_UA_TYPE_HYBRID_LIFT	VTOL (Vertical Take-Off and Landing). Fixed wing aircraft that can take off vertically.
5	MAV_ODID_UA_TYPE_ORNITHOPTER	Ornithopter.
6	MAV_ODID_UA_TYPE_GLIDER	Glider.
7	MAV_ODID_UA_TYPE_KITE	Kite.
8	MAV_ODID_UA_TYPE_FREE_BALLOON	Free Balloon.
9	MAV_ODID_UA_TYPE_CAPTIVE_BALLOON	Captive Balloon.
10	MAV_ODID_UA_TYPE_AIRSHIP	Airship. E.g. a blimp.
11	MAV_ODID_UA_TYPE_FREE_FALL_PARACHUTE	Free Fall/Parachute (unpowered).
12	MAV_ODID_UA_TYPE_ROCKET	Rocket.
13	MAV_ODID_UA_TYPE_TETHERED_POWERED_AIRCRAFT	Tethered powered aircraft.
14	MAV_ODID_UA_TYPE_GROUND_OBSTACLE	Ground Obstacle.

15	MAV_ODID_UA_TYPE_OTHER	Other type of aircraft not listed earlier.

MAV_ODID_STATUS

[Enum]

Value	Field Name	Description
0	MAV_ODID_STATUS_UNDECLARED	The status of the (UA) Unmanned Aircraft is undefined.
1	MAV_ODID_STATUS_GROUND	The UA is on the ground.
2	MAV_ODID_STATUS_AIRBORNE	The UA is in the air.
3	MAV_ODID_STATUS_EMERGENCY	The UA is having an emergency.
4	MAV_ODID_STATUS_REMOTE_ID_SYSTEM_FAILURE	The remote ID system is failing or unreliable in some way.

MAV_ODID_HEIGHT_REF

[Enum]

Value	Field Name	Description
0	MAV_ODID_HEIGHT_REF_OVER_TAKEOFF	The height field is relative to the take-off location.
1	MAV_ODID_HEIGHT_REF_OVER_GROUND	The height field is relative to ground.

MAV_ODID_HOR_ACC

[Enum]

Value	Field Name	Description
0	MAV_ODID_HOR_ACC_UNKNOWN	The horizontal accuracy is unknown.
1	MAV_ODID_HOR_ACC_10NM	The horizontal accuracy is smaller than 10 Nautical Miles. 18.52 km.
2	MAV_ODID_HOR_ACC_4NM	The horizontal accuracy is smaller than 4 Nautical Miles. 7.408 km.
3	MAV_ODID_HOR_ACC_2NM	The horizontal accuracy is smaller than 2 Nautical Miles. 3.704 km.
4	MAV_ODID_HOR_ACC_1NM	The horizontal accuracy is smaller than 1 Nautical Miles. 1.852 km.
5	MAV_ODID_HOR_ACC_0_5NM	The horizontal accuracy is smaller than 0.5 Nautical Miles. 926 m.
6	MAV_ODID_HOR_ACC_0_3NM	The horizontal accuracy is smaller than 0.3 Nautical Miles. 555.6 m.
7	MAV_ODID_HOR_ACC_0_1NM	The horizontal accuracy is smaller than 0.1 Nautical Miles. 185.2 m.
8	MAV_ODID_HOR_ACC_0_05NM	The horizontal accuracy is smaller than 0.05 Nautical Miles. 92.6 m.
9	MAV_ODID_HOR_ACC_30_METER	The horizontal accuracy is smaller than 30 meter.
10	MAV_ODID_HOR_ACC_10_METER	The horizontal accuracy is smaller than 10 meter.
11	MAV_ODID_HOR_ACC_3_METER	The horizontal accuracy is smaller than 3 meter.
12	MAV_ODID_HOR_ACC_1_METER	The horizontal accuracy is smaller than 1 meter.

MAV_ODID_VER_ACC

[Enum]

Value	Field Name	Description
0	MAV_ODID_VER_ACC_UNKNOWN	The vertical accuracy is unknown.
1	MAV_ODID_VER_ACC_150_METER	The vertical accuracy is smaller than 150 meter.
2	MAV_ODID_VER_ACC_45_METER	The vertical accuracy is smaller than 45 meter.
3	MAV_ODID_VER_ACC_25_METER	The vertical accuracy is smaller than 25 meter.
4	MAV_ODID_VER_ACC_10_METER	The vertical accuracy is smaller than 10 meter.
5	MAV_ODID_VER_ACC_3_METER	The vertical accuracy is smaller than 3 meter.
6	MAV_ODID_VER_ACC_1_METER	The vertical accuracy is smaller than 1 meter.

MAV_ODID_SPEED_ACC

[Enum]

Value	Field Name	Description
0	MAV_ODID_SPEED_ACC_UNKNOWN	The speed accuracy is unknown.
1	MAV_ODID_SPEED_ACC_10_METERS_PER_SECOND	The speed accuracy is smaller than 10 meters per second.
2	MAV_ODID_SPEED_ACC_3_METERS_PER_SECOND	The speed accuracy is smaller than 3 meters per second.
3	MAV_ODID_SPEED_ACC_1_METERS_PER_SECOND	The speed accuracy is smaller than 1 meters per second.
4	MAV_ODID_SPEED_ACC_0_3_METERS_PER_SECOND	The speed accuracy is smaller than 0.3 meters per second.

MAV_ODID_TIME_ACC

[Enum]

Value	Field Name	Description
0	MAV_ODID_TIME_ACC_UNKNOWN	The timestamp accuracy is unknown.
1	MAV_ODID_TIME_ACC_0_1_SECOND	The timestamp accuracy is smaller than or equal to 0.1 second.
2	MAV_ODID_TIME_ACC_0_2_SECOND	The timestamp accuracy is smaller than or equal to 0.2 second.
3	MAV_ODID_TIME_ACC_0_3_SECOND	The timestamp accuracy is smaller than or equal to 0.3 second.
4	MAV_ODID_TIME_ACC_0_4_SECOND	The timestamp accuracy is smaller than or equal to 0.4 second.
5	MAV_ODID_TIME_ACC_0_5_SECOND	The timestamp accuracy is smaller than or equal to 0.5 second.
6	MAV_ODID_TIME_ACC_0_6_SECOND	The timestamp accuracy is smaller than or equal to 0.6 second.
7	MAV_ODID_TIME_ACC_0_7_SECOND	The timestamp accuracy is smaller than or equal to 0.7 second.
8	MAV_ODID_TIME_ACC_0_8_SECOND	The timestamp accuracy is smaller than or equal to 0.8 second.
9	MAV_ODID_TIME_ACC_0_9_SECOND	The timestamp accuracy is smaller than or equal to 0.9 second.
10	MAV_ODID_TIME_ACC_1_0_SECOND	The timestamp accuracy is smaller than or equal to 1.0 second.
11	MAV_ODID_TIME_ACC_1_1_SECOND	The timestamp accuracy is smaller than or equal to 1.1 second.
12	MAV_ODID_TIME_ACC_1_2_SECOND	The timestamp accuracy is smaller than or equal to 1.2 second.
13	MAV_ODID_TIME_ACC_1_3_SECOND	The timestamp accuracy is smaller than or equal to 1.3 second.

14	MAV_ODID_TIME_ACC_1_4_SECOND	The timestamp accuracy is smaller than or equal to 1.4 second.
15	MAV_ODID_TIME_ACC_1_5_SECOND	The timestamp accuracy is smaller than or equal to 1.5 second.

MAV_ODID_AUTH_TYPE

[Enum]

Value	Field Name	Description
0	MAV_ODID_AUTH_TYPE_NONE	No authentication type is specified.
1	MAV_ODID_AUTH_TYPE_UAS_ID_SIGNATURE	Signature for the UAS (Unmanned Aircraft System) ID.
2	MAV_ODID_AUTH_TYPE_OPERATOR_ID_SIGNATURE	Signature for the Operator ID.
3	MAV_ODID_AUTH_TYPE_MESSAGE_SET_SIGNATURE	Signature for the entire message set.
4	MAV_ODID_AUTH_TYPE_NETWORK_REMOTE_ID	Authentication is provided by Network Remote ID.
5	MAV_ODID_AUTH_TYPE_SPECIFIC_AUTHENTICATION	The exact authentication type is indicated by the first byte of authentication_data and these type values are managed by ICAO.

MAV_ODID_DESC_TYPE

[Enum]

Value	Field Name	Description
0	MAV_ODID_DESC_TYPE_TEXT	Optional free-form text description of the purpose of the flight.
1	MAV_ODID_DESC_TYPE_EMERGENCY	Optional additional clarification when status == MAV_ODID_STATUS_EMERGENCY.
2	MAV_ODID_DESC_TYPE_EXTENDED_STATUS	Optional additional clarification when status != MAV_ODID_STATUS_EMERGENCY.

MAV_ODID_OPERATOR_LOCATION_TYPE

[Enum]

Value	Field Name	Description
0	MAV_ODID_OPERATOR_LOCATION_TYPE_TAKEOFF	The location/altitude of the operator is the same as the take-off location.
1	MAV_ODID_OPERATOR_LOCATION_TYPE_LIVE_GNSS	The location/altitude of the operator is dynamic. E.g. based on live GNSS data.
2	MAV_ODID_OPERATOR_LOCATION_TYPE_FIXED	The location/altitude of the operator are fixed values.

MAV_ODID_CLASSIFICATION_TYPE

[Enum]

Value	Field Name	Description
0	MAV_ODID_CLASSIFICATION_TYPE_UNDECLARED	The classification type for the UA is undeclared.
1	MAV_ODID_CLASSIFICATION_TYPE_EU	The classification type for the UA follows EU (European Union) specifications.

MAV_ODID_CATEGORY_EU

[Enum]

Value	Field Name	Description
0	MAV_ODID_CATEGORY_EU_UNDECLARED	The category for the UA, according to the EU specification, is undeclared.
1	MAV_ODID_CATEGORY_EU_OPEN	The category for the UA, according to the EU specification, is the Open category.
2	MAV_ODID_CATEGORY_EU_SPECIFIC	The category for the UA, according to the EU specification, is the Specific category.
3	MAV_ODID_CATEGORY_EU_CERTIFIED	The category for the UA, according to the EU specification, is the Certified category.

MAV_ODID_CLASS_EU

[Enum]

Value	Field Name	Description
0	MAV_ODID_CLASS_EU_UNDECLARED	The class for the UA, according to the EU specification, is undeclared.
1	MAV_ODID_CLASS_EU_CLASS_0	The class for the UA, according to the EU specification, is Class 0.
2	MAV_ODID_CLASS_EU_CLASS_1	The class for the UA, according to the EU specification, is Class 1.
3	MAV_ODID_CLASS_EU_CLASS_2	The class for the UA, according to the EU specification, is Class 2.
4	MAV_ODID_CLASS_EU_CLASS_3	The class for the UA, according to the EU specification, is Class 3.
5	MAV_ODID_CLASS_EU_CLASS_4	The class for the UA, according to the EU specification, is Class 4.
6	MAV_ODID_CLASS_EU_CLASS_5	The class for the UA, according to the EU specification, is Class 5.
7	MAV_ODID_CLASS_EU_CLASS_6	The class for the UA, according to the EU specification, is Class 6.

MAV_ODID_OPERATOR_ID_TYPE

[Enum]

Value	Field Name	Description
0	MAV_ODID_OPERATOR_ID_TYPE_CAA	CAA (Civil Aviation Authority) registered operator ID.

TUNE_FORMAT

[Enum] Tune formats (used for vehicle buzzer/tone generation).

Value	Field Name	Description
1	TUNE_FORMAT_QBASIC1_1	Format is QBasic 1.1 Play: https://www.qbasic.net/en/reference/qb11/Statement/PLAY-006.htm.
2	TUNE_FORMAT_MML_MODERN	Format is Modern Music Markup Language (MML): https://en.wikipedia.org/wiki/Music_Macro_Language#Modern_MML.

AIS_TYPE

[Enum] Type of AIS vessel, enum duplicated from AIS standard, https://gpsd.gitlab.io/gpsd/AIVDM.html

Value	Field Name	Description
0	AIS_TYPE_UNKNOWN	Not available (default).
1	AIS_TYPE_RESERVED_1	
2	AIS_TYPE_RESERVED_2	
3	AIS_TYPE_RESERVED_3	
4	AIS_TYPE_RESERVED_4	
5	AIS_TYPE_RESERVED_5	
6	AIS_TYPE_RESERVED_6	
7	AIS_TYPE_RESERVED_7	
8	AIS_TYPE_RESERVED_8	
9	AIS_TYPE_RESERVED_9	
10	AIS_TYPE_RESERVED_10	
11	AIS_TYPE_RESERVED_11	
12	AIS_TYPE_RESERVED_12	
13	AIS_TYPE_RESERVED_13	
14	AIS_TYPE_RESERVED_14	
15	AIS_TYPE_RESERVED_15	
16	AIS_TYPE_RESERVED_16	
17	AIS_TYPE_RESERVED_17	
18	AIS_TYPE_RESERVED_18	
19	AIS_TYPE_RESERVED_19	
20	AIS_TYPE_WIG	Wing In Ground effect.
21	AIS_TYPE_WIG_HAZARDOUS_A	
22	AIS_TYPE_WIG_HAZARDOUS_B	
23	AIS_TYPE_WIG_HAZARDOUS_C	
24	AIS_TYPE_WIG_HAZARDOUS_D	
25	AIS_TYPE_WIG_RESERVED_1	
26	AIS_TYPE_WIG_RESERVED_2	
27	AIS_TYPE_WIG_RESERVED_3	
28	AIS_TYPE_WIG_RESERVED_4	
29	AIS_TYPE_WIG_RESERVED_5	
30	AIS_TYPE_FISHING	
31	AIS_TYPE_TOWING	
32	AIS_TYPE_TOWING_LARGE	Towing: length exceeds 200m or breadth exceeds 25m.
33	AIS_TYPE_DREDGING	Dredging or other underwater ops.
34	AIS_TYPE_DIVING	
35	AIS_TYPE_MILITARY	
36	AIS_TYPE_SAILING	
37	AIS_TYPE_PLEASURE	
38	AIS_TYPE_RESERVED_20	
39	AIS_TYPE_RESERVED_21	
40	AIS_TYPE_HSC	High Speed Craft.
41	AIS_TYPE_HSC_HAZARDOUS_A	
42	AIS_TYPE_HSC_HAZARDOUS_B	
43	AIS_TYPE_HSC_HAZARDOUS_C	
44	AIS_TYPE_HSC_HAZARDOUS_D	
45	AIS_TYPE_HSC_RESERVED_1	
46	AIS_TYPE_HSC_RESERVED_2	
47	AIS_TYPE_HSC_RESERVED_3	
48	AIS_TYPE_HSC_RESERVED_4	
49	AIS_TYPE_HSC_UNKNOWN	
50	AIS_TYPE_PILOT	
51	AIS_TYPE_SAR	Search And Rescue vessel.
52	AIS_TYPE_TUG	
53	AIS_TYPE_PORT_TENDER	
54	AIS_TYPE_ANTI_POLLUTION	Anti-pollution equipment.
55	AIS_TYPE_LAW_ENFORCEMENT	
56	AIS_TYPE_SPARE_LOCAL_1	
57	AIS_TYPE_SPARE_LOCAL_2	
58	AIS_TYPE_MEDICAL_TRANSPORT	
59	AIS_TYPE_NONECOMBATANT	Noncombatant ship according to RR Resolution No. 18.
60	AIS_TYPE_PASSENGER	
61	AIS_TYPE_PASSENGER_HAZARDOUS_A	
62	AIS_TYPE_PASSENGER_HAZARDOUS_B	
63	AIS_TYPE_AIS_TYPE_PASSENGER_HAZARDOUS_C	
64	AIS_TYPE_PASSENGER_HAZARDOUS_D	
65	AIS_TYPE_PASSENGER_RESERVED_1	
66	AIS_TYPE_PASSENGER_RESERVED_2	
67	AIS_TYPE_PASSENGER_RESERVED_3	
68	AIS_TYPE_AIS_TYPE_PASSENGER_RESERVED_4	
69	AIS_TYPE_PASSENGER_UNKNOWN	
70	AIS_TYPE_CARGO	
71	AIS_TYPE_CARGO_HAZARDOUS_A	
72	AIS_TYPE_CARGO_HAZARDOUS_B	
73	AIS_TYPE_CARGO_HAZARDOUS_C	

74	AIS_TYPE_CARGO_HAZARDOUS_D
75	AIS_TYPE_CARGO_RESERVED_1
76	AIS_TYPE_CARGO_RESERVED_2
77	AIS_TYPE_CARGO_RESERVED_3
78	AIS_TYPE_CARGO_RESERVED_4
79	AIS_TYPE_CARGO_UNKNOWN
80	AIS_TYPE_TANKER
81	AIS_TYPE_TANKER_HAZARDOUS_A
82	AIS_TYPE_TANKER_HAZARDOUS_B
83	AIS_TYPE_TANKER_HAZARDOUS_C
84	AIS_TYPE_TANKER_HAZARDOUS_D
85	AIS_TYPE_TANKER_RESERVED_1
86	AIS_TYPE_TANKER_RESERVED_2
87	AIS_TYPE_TANKER_RESERVED_3
88	AIS_TYPE_TANKER_RESERVED_4
89	AIS_TYPE_TANKER_UNKNOWN
90	AIS_TYPE_OTHER
91	AIS_TYPE_OTHER_HAZARDOUS_A
92	AIS_TYPE_OTHER_HAZARDOUS_B
93	AIS_TYPE_OTHER_HAZARDOUS_C
94	AIS_TYPE_OTHER_HAZARDOUS_D
95	AIS_TYPE_OTHER_RESERVED_1
96	AIS_TYPE_OTHER_RESERVED_2
97	AIS_TYPE_OTHER_RESERVED_3
98	AIS_TYPE_OTHER_RESERVED_4
99	AIS_TYPE_OTHER_UNKNOWN

AIS_NAV_STATUS

[Enum] Navigational status of AIS vessel, enum duplicated from AIS standard, https://gpsd.gitlab.io/gpsd/AIVDM.html

Value	Field Name	Description
0	UNDER_WAY	Under way using engine.
1	AIS_NAV_ANCHORED	
2	AIS_NAV_UN_COMMANDED	
3	AIS_NAV_RESTRICTED_MANOEUVERABILITY	
4	AIS_NAV_DRAUGHT_CONSTRAINED	
5	AIS_NAV_MOORED	
6	AIS_NAV_AGROUND	
7	AIS_NAV_FISHING	
8	AIS_NAV_SAILING	
9	AIS_NAV_RESERVED_HSC	
10	AIS_NAV_RESERVED_WIG	
11	AIS_NAV_RESERVED_1	
12	AIS_NAV_RESERVED_2	
13	AIS_NAV_RESERVED_3	
14	AIS_NAV_AIS_SART	Search And Rescue Transponder.
15	AIS_NAV_UNKNOWN	Not available (default).

AIS_FLAGS

[Enum] These flags are used in the AIS_VESSEL.fields bitmask to indicate validity of data in the other message fields. When set, the data is valid.

Value	Field Name	Description
1	AIS_FLAGS_POSITION_ACCURACY	1 = Position accuracy less than 10m, 0 = position accuracy greater than 10m.
2	AIS_FLAGS_VALID_COG	
4	AIS_FLAGS_VALID_VELOCITY	
8	AIS_FLAGS_HIGH_VELOCITY	1 = Velocity over 52.5765m/s (102.2 knots)
16	AIS_FLAGS_VALID_TURN_RATE	
32	AIS_FLAGS_TURN_RATE_SIGN_ONLY	Only the sign of the returned turn rate value is valid, either greater than 5deg/30s or less than -5deg/30s
64	AIS_FLAGS_VALID_DIMENSIONS	
128	AIS_FLAGS_LARGE_BOW_DIMENSION	Distance to bow is larger than 511m
256	AIS_FLAGS_LARGE_STERN_DIMENSION	Distance to stern is larger than 511m
512	AIS_FLAGS_LARGE_PORT_DIMENSION	Distance to port side is larger than 63m
1024	AIS_FLAGS_LARGE_STARBOARD_DIMENSION	Distance to starboard side is larger than 63m
2048	AIS_FLAGS_VALID_CALLSIGN	
4096	AIS_FLAGS_VALID_NAME	

FAILURE_UNIT

[Enum] List of possible units where failures can be injected.

Value	Field Name	Description
0	FAILURE_UNIT_SENSOR_GYRO	
1	FAILURE_UNIT_SENSOR_ACCEL	
2	FAILURE_UNIT_SENSOR_MAG	
3	FAILURE_UNIT_SENSOR_BARO	
4	FAILURE_UNIT_SENSOR_GPS	
5	FAILURE_UNIT_SENSOR_OPTICAL_FLOW	
6	FAILURE_UNIT_SENSOR_VIO	

7	FAILURE_UNIT_SENSOR_DISTANCE_SENSOR
8	FAILURE_UNIT_SENSOR_AIRSPEED
100	FAILURE_UNIT_SYSTEM_BATTERY
101	FAILURE_UNIT_SYSTEM_MOTOR
102	FAILURE_UNIT_SYSTEM_SERVO
103	FAILURE_UNIT_SYSTEM_AVOIDANCE
104	FAILURE_UNIT_SYSTEM_RC_SIGNAL
105	FAILURE_UNIT_SYSTEM_MAVLINK_SIGNAL

FAILURE_TYPE

[Enum] List of possible failure type to inject.

Value	Field Name	Description
0	FAILURE_TYPE_OK	No failure injected, used to reset a previous failure.
1	FAILURE_TYPE_OFF	Sets unit off, so completely non-responsive.
2	FAILURE_TYPE_STUCK	Unit is stuck e.g. keeps reporting the same value.
3	FAILURE_TYPE_GARBAGE	Unit is reporting complete garbage.
4	FAILURE_TYPE_WRONG	Unit is consistently wrong.
5	FAILURE_TYPE_SLOW	Unit is slow, so e.g. reporting at slower than expected rate.
6	FAILURE_TYPE_DELAYED	Data of unit is delayed in time.
7	FAILURE_TYPE_INTERMITTENT	Unit is sometimes working, sometimes not.

NAV_VTOL_LAND_OPTIONS

[Enum]

Value	Field Name	Description
0	NAV_VTOL_LAND_OPTIONS_DEFAULT	Default autopilot landing behaviour.
1	NAV_VTOL_LAND_OPTIONS_FW_DESCENT	Descend in fixed wing mode, transitioning to multicopter mode for vertical landing when close to the ground. The fixed wing descent pattern is at the discretion of the vehicle (e.g. transition altitude, loiter direction, radius, and speed, etc.).
2	NAV_VTOL_LAND_OPTIONS_HOVER_DESCENT	Land in multicopter mode on reaching the landing coordinates (the whole landing is by "hover descent").

MAV_WINCH_STATUS_FLAG

[Enum] Winch status flags used in WINCH_STATUS

Value	Field Name	Description
1	MAV_WINCH_STATUS_HEALTHY	Winch is healthy
2	MAV_WINCH_STATUS_FULLY_RETRACTED	Winch line is fully retracted
4	MAV_WINCH_STATUS_MOVING	Winch motor is moving
8	MAV_WINCH_STATUS_CLUTCH_ENGAGED	Winch clutch is engaged allowing motor to move freely.
16	MAV_WINCH_STATUS_LOCKED	Winch is locked by locking mechanism.
32	MAV_WINCH_STATUS_DROPPING	Winch is gravity dropping payload.
64	MAV_WINCH_STATUS_ARRESTING	Winch is arresting payload descent.
128	MAV_WINCH_STATUS_GROUND_SENSE	Winch is using torque measurements to sense the ground.
256	MAV_WINCH_STATUS_RETRACTING	Winch is returning to the fully retracted position.
512	MAV_WINCH_STATUS_REDELIVER	Winch is redelivering the payload. This is a failover state if the line tension goes above a threshold during RETRACTING.
1024	MAV_WINCH_STATUS_ABANDON_LINE	Winch is abandoning the line and possibly payload. Winch unspools the entire calculated line length. This is a failover state from REDELIVER if the number of attempts exceeds a threshold.

MAG_CAL_STATUS

[Enum]

Value	Field Name	Description
0	MAG_CAL_NOT_STARTED	
1	MAG_CAL_WAITING_TO_START	
2	MAG_CAL_RUNNING_STEP_ONE	
3	MAG_CAL_RUNNING_STEP_TWO	
4	MAG_CAL_SUCCESS	
5	MAG_CAL_FAILED	
6	MAG_CAL_BAD_ORIENTATION	
7	MAG_CAL_BAD_RADIUS	

MAV_EVENT_ERROR_REASON

[Enum] Reason for an event error response.

Value	Field Name	Description
0	MAV_EVENT_ERROR_REASON_UNAVAILABLE	The requested event is not available (anymore).

MAV_EVENT_CURRENT_SEQUENCE_FLAGS

[Enum] Flags for CURRENT_EVENT_SEQUENCE.

Value	Field Name	Description
1	MAV_EVENT_CURRENT_SEQUENCE_FLAGS_RESET	A sequence reset has happened (e.g. vehicle reboot).

HIL_SENSOR_UPDATED_FLAGS

[Enum] Flags in the HIL_SENSOR message indicate which fields have updated since the last message

Value	Field Name	Description
0	HIL_SENSOR_UPDATED_NONE	None of the fields in HIL_SENSOR have been updated
1	HIL_SENSOR_UPDATED_XACC	The value in the xacc field has been updated
2	HIL_SENSOR_UPDATED_YACC	The value in the yacc field has been updated
4	HIL_SENSOR_UPDATED_ZACC	The value in the zacc field has been updated
8	HIL_SENSOR_UPDATED_XGYRO	The value in the xgyro field has been updated
16	HIL_SENSOR_UPDATED_YGYRO	The value in the ygyro field has been updated
32	HIL_SENSOR_UPDATED_ZGYRO	The value in the zgyro field has been updated
64	HIL_SENSOR_UPDATED_XMAG	The value in the xmag field has been updated
128	HIL_SENSOR_UPDATED_YMAG	The value in the ymag field has been updated
256	HIL_SENSOR_UPDATED_ZMAG	The value in the zmag field has been updated
512	HIL_SENSOR_UPDATED_ABS_PRESSURE	The value in the abs_pressure field has been updated
1024	HIL_SENSOR_UPDATED_DIFF_PRESSURE	The value in the diff_pressure field has been updated
2048	HIL_SENSOR_UPDATED_PRESSURE_ALT	The value in the pressure_alt field has been updated
4096	HIL_SENSOR_UPDATED_TEMPERATURE	The value in the temperature field has been updated

HIGHRES_IMU_UPDATED_FLAGS

[Enum] Flags in the HIGHRES_IMU message indicate which fields have updated since the last message

Value	Field Name	Description
0	HIGHRES_IMU_UPDATED_NONE	None of the fields in HIGHRES_IMU have been updated
1	HIGHRES_IMU_UPDATED_XACC	The value in the xacc field has been updated
2	HIGHRES_IMU_UPDATED_YACC	The value in the yacc field has been updated
4	HIGHRES_IMU_UPDATED_ZACC	The value in the zacc field has been updated since
8	HIGHRES_IMU_UPDATED_XGYRO	The value in the xgyro field has been updated
16	HIGHRES_IMU_UPDATED_YGYRO	The value in the ygyro field has been updated
32	HIGHRES_IMU_UPDATED_ZGYRO	The value in the zgyro field has been updated
64	HIGHRES_IMU_UPDATED_XMAG	The value in the xmag field has been updated
128	HIGHRES_IMU_UPDATED_YMAG	The value in the ymag field has been updated
256	HIGHRES_IMU_UPDATED_ZMAG	The value in the zmag field has been updated
512	HIGHRES_IMU_UPDATED_ABS_PRESSURE	The value in the abs_pressure field has been updated
1024	HIGHRES_IMU_UPDATED_DIFF_PRESSURE	The value in the diff_pressure field has been updated
2048	HIGHRES_IMU_UPDATED_PRESSURE_ALT	The value in the pressure_alt field has been updated
4096	HIGHRES_IMU_UPDATED_TEMPERATURE	The value in the temperature field has been updated
65535	HIGHRES_IMU_UPDATED_ALL	All fields in HIGHRES_IMU have been updated.

CAN_FILTER_OP

[Enum]

Value	Field Name	Description
0	CAN_FILTER_REPLACE	
1	CAN_FILTER_ADD	
2	CAN_FILTER_REMOVE	

MAV_FTP_ERR

[Enum] MAV FTP error codes (https://mavlink.io/en/services/ftp.html)

Value	Field Name	Description
0	MAV_FTP_ERR_NONE	None: No error
1	MAV_FTP_ERR_FAIL	Fail: Unknown failure
2	MAV_FTP_ERR_FAILERRNO	FailErrno: Command failed, Err number sent back in PayloadHeader.data[1]. This is a file-system error number understood by the server operating system.
3	MAV_FTP_ERR_INVALIDDATASIZE	InvalidDataSize: Payload size is invalid
4	MAV_FTP_ERR_INVALIDSESSION	InvalidSession: Session is not currently open
5	MAV_FTP_ERR_NOSESSIONSAVAILABLE	NoSessionsAvailable: All available sessions are already in use
6	MAV_FTP_ERR_EOF	EOF: Offset past end of file for ListDirectory and ReadFile commands
7	MAV_FTP_ERR_UNKNOWNCOMMAND	UnknownCommand: Unknown command / opcode
8	MAV_FTP_ERR_FILEEXISTS	FileExists: File/directory already exists
9	MAV_FTP_ERR_FILEPROTECTED	FileProtected: File/directory is write protected
10	MAV_FTP_ERR_FILENOTFOUND	FileNotFound: File/directory not found

MAV_FTP_OPCODE

[Enum] MAV FTP opcodes: https://mavlink.io/en/services/ftp.html

Value	Field Name	Description
0	MAV_FTP_OPCODE_NONE	None. Ignored, always ACKed
1	MAV_FTP_OPCODE_TERMINATESESSION	TerminateSession: Terminates open Read session
2	MAV_FTP_OPCODE_RESETSESSION	ResetSessions: Terminates all open read sessions
3	MAV_FTP_OPCODE_LISTDIRECTORY	ListDirectory. List files and directories in path from offset
4	MAV_FTP_OPCODE_OPENFILERO	OpenFileRO: Opens file at path for reading, returns session
5	MAV_FTP_OPCODE_READFILE	ReadFile: Reads size bytes from offset in session
6	MAV_FTP_OPCODE_CREATEFILE	CreateFile: Creates file at path for writing, returns session
7	MAV_FTP_OPCODE_WRITEFILE	WriteFile: Writes size bytes to offset in session
8	MAV_FTP_OPCODE_REMOVEFILE	RemoveFile: Remove file at path
9	MAV_FTP_OPCODE_CREATEDIRECTORY	CreateDirectory: Creates directory at path
10	MAV_FTP_OPCODE_REMOVEDIRECTORY	RemoveDirectory: Removes directory at path. The directory must be empty.
11	MAV_FTP_OPCODE_OPENFILEWO	OpenFileWO: Opens file at path for writing, returns session
12	MAV_FTP_OPCODE_TRUNCATEFILE	TruncateFile: Truncate file at path to offset length
13	MAV_FTP_OPCODE_RENAME	Rename: Rename path1 to path2
14	MAV_FTP_OPCODE_CALCFILECRC	CalcFileCRC32: Calculate CRC32 for file at path
15	MAV_FTP_OPCODE_BURSTREADFILE	BurstReadFile: Burst download session file
128	MAV_FTP_OPCODE_ACK	ACK: ACK response
129	MAV_FTP_OPCODE_NAK	NAK: NAK response

MAVLink Commands (MAV_CMD)

> MAVLink commands (MAV_CMD) and messages are different! These commands define the values of up to 7 parameters that are packaged INSIDE specific messages used in the Mission Protocol and Command Protocol. Use commands for actions in missions or if you need acknowledgment and/or retry logic from a request. Otherwise use messages.

Commands to be executed by the MAV. They can be executed on user request, or as part of a mission script. If the action is used in a mission, the parameter mapping to the waypoint/mission message is as follows: Param 1, Param 2, Param 3, Param 4, X: Param 5, Y:Param 6, Z:Param 7. This command list is similar what ARINC 424 is for commercial aircraft: A data format how to interpret waypoint/mission data. NaN and INT32_MAX may be used in float/integer params (respectively) to indicate optional/default values (e.g. to use the component's current yaw or latitude rather than a specific value). See https://mavlink.io/en/guide/xml_schema.html#MAV_CMD for information about the structure of the MAV_CMD entries

MAV_CMD_NAV_WAYPOINT (16

)

[Command] Navigate to waypoint.

Param (:Label)	Description	Values	Units
1: Hold	Hold time. (ignored by fixed wing, time to stay at waypoint for rotary wing)	*min:*0	s
2: Accept Radius	Acceptance radius (if the sphere with this radius is hit, the waypoint counts as reached)	*min:*0	m
3: Pass Radius	0 to pass through the WP, if > 0 radius to pass by WP. Positive value for clockwise orbit, negative value for counter-clockwise orbit. Allows trajectory control.		m
4: Yaw	Desired yaw angle at waypoint (rotary wing). NaN to use the current system yaw heading mode (e.g. yaw towards next waypoint, yaw to home, etc.).		deg
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

MAV_CMD_NAV_LOITER_UNLIM (17

)

[Command] Loiter around this waypoint an unlimited amount of time

Param (:Label)	Description	Units
1	Empty	
2	Empty	
3: Radius	Loiter radius around waypoint for forward-only moving vehicles (not multicopters). If positive loiter clockwise, else counter-clockwise	m
4: Yaw	Desired yaw angle. NaN to use the current system yaw heading mode (e.g. yaw towards next waypoint, yaw to home, etc.).	deg
5: Latitude	Latitude	
6: Longitude	Longitude	
7: Altitude	Altitude	m

MAV_CMD_NAV_LOITER_TURNS (18

)

[Command] Loiter around this waypoint for X turns

Param (:Label)	Description	Values	Units
1: Turns	Number of turns.	*min:*0	
2: Heading Required	Leave loiter circle only once heading towards the next waypoint (0 = False)	*min:*0 *max:*1 *increment:*1	
3: Radius	Loiter radius around waypoint for forward-only moving vehicles (not multicopters). If positive loiter clockwise, else counter-clockwise		m

4: Xtrack Location	Loiter circle exit location and/or path to next waypoint ("xtrack") for forward-only moving vehicles (not multicopters). 0 for the vehicle to converge towards the center xtrack when it leaves the loiter (the line between the centers of the current and next waypoint), 1 to converge to the direct line between the location that the vehicle exits the loiter radius and the next waypoint. Otherwise the angle (in degrees) between the tangent of the loiter circle and the center xtrack at which the vehicle must leave the loiter (and converge to the center xtrack). NaN to use the current system default xtrack behaviour.		
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

MAV_CMD_NAV_LOITER_TIME (19

)

[Command] Loiter at the specified latitude, longitude and altitude for a certain amount of time. Multicopter vehicles stop at the point (within a vehicle-specific acceptance radius). Forward-only moving vehicles (e.g. fixed-wing) circle the point with the specified radius/direction. If the Heading Required parameter (2) is non-zero forward moving aircraft will only leave the loiter circle once heading towards the next waypoint.

Param (:Label)	Description	Values	Units
1: Time	Loiter time (only starts once Lat, Lon and Alt is reached).	*min:*0	s
2: Heading Required	Leave loiter circle only once heading towards the next waypoint (0 = False)	*min:*0 *max:*1 *increment:*1	
3: Radius	Loiter radius around waypoint for forward-only moving vehicles (not multicopters). If positive loiter clockwise, else counter-clockwise.		m
4: Xtrack Location	Loiter circle exit location and/or path to next waypoint ("xtrack") for forward-only moving vehicles (not multicopters). 0 for the vehicle to converge towards the center xtrack when it leaves the loiter (the line between the centers of the current and next waypoint), 1 to converge to the direct line between the location that the vehicle exits the loiter radius and the next waypoint. Otherwise the angle (in degrees) between the tangent of the loiter circle and the center xtrack at which the vehicle must leave the loiter (and converge to the center xtrack). NaN to use the current system default xtrack behaviour.		
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

MAV_CMD_NAV_RETURN_TO_LAUNCH (20

)

[Command] Return to launch location

Param (:Label)	Description
1	Empty
2	Empty
3	Empty
4	Empty
5	Empty
6	Empty
7	Empty

MAV_CMD_NAV_LAND (21

)

[Command] Land at location.

Param (:Label)	Description	Values	Units
1: Abort Alt	Minimum target altitude if landing is aborted (0 = undefined/use system default).		m
2: Land Mode	Precision land mode.	PRECISION_LAND_MODE	
3	Empty.		
4: Yaw Angle	Desired yaw angle. NaN to use the current system yaw heading mode (e.g. yaw towards next waypoint, yaw to home, etc.).		deg
5: Latitude	Latitude.		
6: Longitude	Longitude.		
7: Altitude	Landing altitude (ground level in current frame).		m

MAV_CMD_NAV_TAKEOFF (22

)

[Command] Takeoff from ground / hand. Vehicles that support multiple takeoff modes (e.g. VTOL quadplane) should take off using the currently configured mode.

Param (:Label)	Description	Units
1: Pitch	Minimum pitch (if airspeed sensor present), desired pitch without sensor	deg
2	Empty	
3	Empty	
	Yaw angle (if magnetometer present), ignored without magnetometer. NaN to use the current system yaw heading	

4: Yaw	mode (e.g. yaw towards next waypoint, yaw to home, etc.).	deg
5: Latitude	Latitude	
6: Longitude	Longitude	
7: Altitude	Altitude	m

MAV_CMD_NAV_LAND_LOCAL (23

)

[Command] Land at local position (local frame only)

Param (:Label)	Description	Values	Units
1: Target	Landing target number (if available)	*min:*0 *increment:*1	
2: Offset	Maximum accepted offset from desired landing position - computed magnitude from spherical coordinates: d = sqrt(x^2 + y^2 + z^2), which gives the maximum accepted distance between the desired landing position and the position where the vehicle is about to land	*min:*0	m
3: Descend Rate	Landing descend rate		m/s
4: Yaw	Desired yaw angle		rad
5: Y Position	Y-axis position		m
6: X Position	X-axis position		m
7: Z Position	Z-axis / ground level position		m

MAV_CMD_NAV_TAKEOFF_LOCAL (24

)

[Command] Takeoff from local position (local frame only)

Param (:Label)	Description	Units
1: Pitch	Minimum pitch (if airspeed sensor present), desired pitch without sensor	rad
2	Empty	
3: Ascend Rate	Takeoff ascend rate	m/s
4: Yaw	Yaw angle (if magnetometer or another yaw estimation source present), ignored without one of these	rad
5: Y Position	Y-axis position	m
6: X Position	X-axis position	m
7: Z Position	Z-axis position	m

MAV_CMD_NAV_FOLLOW (25

)

[Command] Vehicle following, i.e. this waypoint represents the position of a moving vehicle

Param (:Label)	Description	Values	Units
1: Following	Following logic to use (e.g. loitering or sinusoidal following) - depends on specific autopilot implementation	*increment:*1	
2: Ground Speed	Ground speed of vehicle to be followed		m/s
3: Radius	Radius around waypoint. If positive loiter clockwise, else counter-clockwise		m
4: Yaw	Desired yaw angle.		deg
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

MAV_CMD_NAV_CONTINUE_AND_CHANGE_ALT (30

)

[Command] Continue on the current course and climb/descend to specified altitude. When the altitude is reached continue to the next command (i.e., don't proceed to the next command until the desired altitude is reached.

Param (:Label)	Description	Values	Units
1: Action	Climb or Descend (0 = Neutral, command completes when within 5m of this command's altitude, 1 = Climbing, command completes when at or above this command's altitude, 2 = Descending, command completes when at or below this command's altitude.	*min:*0 *max:*2 *increment:*1	
2	Empty		
3	Empty		
4	Empty		
5	Empty		
6	Empty		
7: Altitude	Desired altitude		m

MAV_CMD_NAV_LOITER_TO_ALT (31

)

[Command] Begin loiter at the specified Latitude and Longitude. If Lat=Lon=0, then loiter at the current position. Don't consider the navigation command complete (don't leave loiter) until the altitude has been reached. Additionally, if the Heading Required parameter is non-zero the aircraft will not leave the loiter until heading toward the next waypoint.

Param (:Label)	Description	Values	Units
1: Heading Required	Leave loiter circle only once heading towards the next waypoint (0 = False)	*min:*0 *max:*1 *increment:*1	
2: Radius	Loiter radius around waypoint for forward-only moving vehicles (not multicopters). If positive loiter clockwise, negative counter-clockwise, 0 means no change to standard loiter.		m
3	Empty		
4: Xtrack Location	Loiter circle exit location and/or path to next waypoint ("xtrack") for forward-only moving vehicles (not multicopters). 0 for the vehicle to converge towards the center xtrack when it leaves the loiter (the line between the centers of the current and next waypoint), 1 to converge to the direct line between the location that the vehicle exits the loiter radius and the next waypoint. Otherwise the angle (in degrees) between the tangent of the loiter circle and the center xtrack at which the vehicle must leave the loiter (and converge to the center xtrack). NaN to use the current system default xtrack behaviour.	*min:*0 *max:*1 *increment:*1	
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

MAV_CMD_DO_FOLLOW (32

)

[Command] Begin following a target

Param (:Label)	Description	Values	Units
1: System ID	System ID (of the FOLLOW_TARGET beacon). Send 0 to disable follow-me and return to the default position hold mode.	*min:*0 *max:*255 *increment:*1	
2	Reserved		
3	Reserved		
4: Altitude Mode	Altitude mode: 0: Keep current altitude, 1: keep altitude difference to target, 2: go to a fixed altitude above home.	*min:*0 *max:*2 *increment:*1	
5: Altitude	Altitude above home. (used if mode=2)		m
6	Reserved		
7: Time to Land	Time to land in which the MAV should go to the default position hold mode after a message RX timeout.	*min:*0	s

MAV_CMD_DO_FOLLOW_REPOSITION (33

)

[Command] Reposition the MAV after a follow target command has been sent

Param (:Label)	Description	Units
1: Camera Q1	Camera q1 (where 0 is on the ray from the camera to the tracking device)	
2: Camera Q2	Camera q2	
3: Camera Q3	Camera q3	
4: Camera Q4	Camera q4	
5: Altitude Offset	altitude offset from target	m
6: X Offset	X offset from target	m
7: Y Offset	Y offset from target	m

MAV_CMD_DO_ORBIT (34

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Command] Start orbiting on the circumference of a circle defined by the parameters. Setting values to NaN/INT32_MAX (as appropriate) results in using defaults.

Param (:Label)	Description	Values	Units
1: Radius	Radius of the circle. Positive: orbit clockwise. Negative: orbit counter-clockwise. NaN: Use vehicle default radius, or current radius if already orbiting.		m
2: Velocity	Tangential Velocity. NaN: Use vehicle default velocity, or current velocity if already orbiting.		m/s
3: Yaw Behavior	Yaw behavior of the vehicle.	ORBIT_YAW_BEHAVIOUR	
4: Orbits	Orbit around the centre point for this many radians (i.e. for a three-quarter orbit set 270*Pi/180). 0: Orbit forever. NaN: Use vehicle default, or current value if already orbiting.	*min:*0	rad
5: Latitude/X	Center point latitude (if no MAV_FRAME specified) / X coordinate according to MAV_FRAME. INT32_MAX (or NaN if sent in COMMAND_LONG): Use current vehicle position, or current center if already orbiting.		
6: Longitude/Y	Center point longitude (if no MAV_FRAME specified) / Y coordinate according to MAV_FRAME. INT32_MAX (or NaN if sent in COMMAND_LONG): Use current vehicle position, or current center if already orbiting.		
7: Altitude/Z	Center point altitude (MSL) (if no MAV_FRAME specified) / Z coordinate according to MAV_FRAME. NaN: Use current vehicle altitude.		

MAV_CMD_NAV_ROI (80

)

DEPRECATED: Replaced by MAV_CMD_DO_SET_ROI_* (2018-01).

[Command] Sets the region of interest (ROI) for a sensor set or the vehicle itself. This can then be used by the vehicle's control system to control the vehicle attitude and the attitude of various sensors such as cameras.

Param (:Label)	Description	Values
1: ROI Mode	Region of interest mode.	MAV_ROI
2: WP Index	Waypoint index/ target ID. (see MAV_ROI enum)	*min:*0 *increment:*1
3: ROI Index	ROI index (allows a vehicle to manage multiple ROI's)	*min:*0 *increment:*1
4	Empty	
5: X	x the location of the fixed ROI (see MAV_FRAME)	
6: Y	y	
7: Z	z	

MAV_CMD_NAV_PATHPLANNING (81

)

[Command] Control autonomous path planning on the MAV.

Param (:Label)	Description	Values	Units
1: Local Ctrl	0: Disable local obstacle avoidance / local path planning (without resetting map), 1: Enable local path planning, 2: Enable and reset local path planning	*min:*0 *max:*2 *increment:*1	
2: Global Ctrl	0: Disable full path planning (without resetting map), 1: Enable, 2: Enable and reset map/occupancy grid, 3: Enable and reset planned route, but not occupancy grid	*min:*0 *max:*3 *increment:*1	
3	Empty		
4: Yaw	Yaw angle at goal		deg
5: Latitude/X	Latitude/X of goal		
6: Longitude/Y	Longitude/Y of goal		
7: Altitude/Z	Altitude/Z of goal		

MAV_CMD_NAV_SPLINE_WAYPOINT (82

)

[Command] Navigate to waypoint using a spline path.

Param (:Label)	Description	Values	Units
1: Hold	Hold time. (ignored by fixed wing, time to stay at waypoint for rotary wing)	*min:*0	s
2	Empty		
3	Empty		
4	Empty		
5: Latitude/X	Latitude/X of goal		
6: Longitude/Y	Longitude/Y of goal		
7: Altitude/Z	Altitude/Z of goal		

MAV_CMD_NAV_VTOL_TAKEOFF (84

)

[Command] Takeoff from ground using VTOL mode, and transition to forward flight with specified heading. The command should be ignored by vehicles that dont support both VTOL and fixed-wing flight (multicopters, boats,etc.).

Param (:Label)	Description	Values	Units
1	Empty		
2: Transition Heading	Front transition heading.	VTOL_TRANSITION_HEADING	
3	Empty		
4: Yaw Angle	Yaw angle. NaN to use the current system yaw heading mode (e.g. yaw towards next waypoint, yaw to home, etc.).		deg
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

MAV_CMD_NAV_VTOL_LAND (85

)

[Command] Land using VTOL mode

Param (:Label)	Description	Values	Units
1: Land Options	Landing behaviour.	NAV_VTOL_LAND_OPTIONS	
2	Empty		
3: Approach Altitude	Approach altitude (with the same reference as the Altitude field). NaN if unspecified.		m
4: Yaw	Yaw angle. NaN to use the current system yaw heading mode (e.g. yaw towards next waypoint, yaw to home, etc.).		deg

5: Latitude	Latitude	
6: Longitude	Longitude	
7: Ground Altitude	Altitude (ground level) relative to the current coordinate frame. NaN to use system default landing altitude (ignore value).	m

MAV_CMD_NAV_GUIDED_ENABLE (92

)

[Command] hand control over to an external controller

Param (:Label)	Description	Values
1: Enable	On / Off (> 0.5f on)	*min:*0 *max:*1 *increment:*1
2	Empty	
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_NAV_DELAY (93

)

[Command] Delay the next navigation command a number of seconds or until a specified time

Param (:Label)	Description	Values	Units
1: Delay	Delay (-1 to enable time-of-day fields)	*min:* -1 *increment:*1	s
2: Hour	hour (24h format, UTC, -1 to ignore)	*min:* -1 *max:*23 *increment:*1	
3: Minute	minute (24h format, UTC, -1 to ignore)	*min:* -1 *max:*59 *increment:*1	
4: Second	second (24h format, UTC, -1 to ignore)	*min:* -1 *max:*59 *increment:*1	
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_NAV_PAYLOAD_PLACE (94

)

[Command] Descend and place payload. Vehicle moves to specified location, descends until it detects a hanging payload has reached the ground, and then releases the payload. If ground is not detected before the reaching the maximum descent value (param1), the command will complete without releasing the payload.

Param (:Label)	Description	Values	Units
1: Max Descent	Maximum distance to descend.	*min:*0	m
2	Empty		
3	Empty		
4	Empty		
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

MAV_CMD_NAV_LAST (95

)

[Command] NOP - This command is only used to mark the upper limit of the NAV/ACTION commands in the enumeration

Param (:Label)	Description
1	Empty
2	Empty
3	Empty
4	Empty
5	Empty
6	Empty
7	Empty

MAV_CMD_CONDITION_DELAY (112

)

[Command] Delay mission state machine.

Param (:Label)	Description	Values	Units
1: Delay	Delay	*min:*0	s
2	Empty		
3	Empty		

4	Empty
5	Empty
6	Empty
7	Empty

MAV_CMD_CONDITION_CHANGE_ALT (113

)

[Command] Ascend/descend to target altitude at specified rate. Delay mission state machine until desired altitude reached.

Param (:Label)	Description	Units
1: Rate	Descent / Ascend rate.	m/s
2	Empty	
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7: Altitude	Target Altitude	m

MAV_CMD_CONDITION_DISTANCE (114

)

[Command] Delay mission state machine until within desired distance of next NAV point.

Param (:Label)	Description	Values	Units
1: Distance	Distance.	*min:*0	m
2	Empty		
3	Empty		
4	Empty		
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_CONDITION_YAW (115

)

[Command] Reach a certain target angle.

Param (:Label)	Description	Values	Units
1: Angle	target angle, 0 is north		deg
2: Angular Speed	angular speed		deg/s
3: Direction	direction: -1: counter clockwise, 1: clockwise *min:* -1 *max:*1 *increment:*2		
4: Relative	0: absolute angle, 1: relative offset	*min:*0 *max:*1 *increment:*1	
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_CONDITION_LAST (159

)

[Command] NOP - This command is only used to mark the upper limit of the CONDITION commands in the enumeration

Param (:Label)	Description
1	Empty
2	Empty
3	Empty
4	Empty
5	Empty
6	Empty
7	Empty

MAV_CMD_DO_SET_MODE (176

)

[Command] Set system mode.

Param (:Label)	Description	Values
1: Mode	Mode	MAV_MODE
2: Custom Mode	Custom mode - this is system specific, please refer to the individual autopilot specifications for details.	
3: Custom Submode	Custom sub mode - this is system specific, please refer to the individual autopilot specifications for details.	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_JUMP (177

)

[Command] Jump to the desired command in the mission list. Repeat this action only the specified number of times

Param (:Label)	Description	Values
1: Number	Sequence number	*min:*0 *increment:*1
2: Repeat	Repeat count	*min:*0 *increment:*1
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_CHANGE_SPEED (178

)

[Command] Change speed and/or throttle set points.

Param (:Label)	Description	Values	Units
1: Speed Type	Speed type (0=Airspeed, 1=Ground Speed, 2=Climb Speed, 3=Descent Speed)	*min:*0 *max:*3 *increment:*1	
2: Speed	Speed (-1 indicates no change)	*min:*	
-1			
m/s			
3: Throttle	Throttle (-1 indicates no change)	*min:*	
-1			

| % | | 4 | Reserved (set to 0) | | | | 5 | Reserved (set to 0) | | | | 6 | Reserved (set to 0) | | | | 7 | Reserved (set to 0) | | |

MAV_CMD_DO_SET_HOME (179

)

[Command] Sets the home position to either to the current position or a specified position. The home position is the default position that the system will return to and land on. The position is set automatically by the system during the takeoff (and may also be set using this command). Note: the current home position may be emitted in a HOME_POSITION message on request (using MAV_CMD_REQUEST_MESSAGE with param1=242).

Param (:Label)	Description	Values	Units
1: Use Current	Use current (1=use current location, 0=use specified location)	*min:*0 *max:*1 *increment:*1	
2	Empty		
3	Empty		
4: Yaw	Yaw angle. NaN to use default heading		deg
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

MAV_CMD_DO_SET_PARAMETER (180

)

[Command] Set a system parameter. Caution! Use of this command requires knowledge of the numeric enumeration value of the parameter.

Param (:Label)	Description	Values
1: Number	Parameter number	*min:*0 *increment:*1
2: Value	Parameter value	
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_SET_RELAY (181

)

[Command] Set a relay to a condition.

Param (:Label)	Description	Values
1: Instance	Relay instance number.	*min:*0 *increment:*1
2: Setting	Setting. (1=on, 0=off, others possible depending on system hardware)	*min:*0 *increment:*1
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_REPEAT_RELAY (182

)

[Command] Cycle a relay on and off for a desired number of cycles with a desired period.

Param (:Label)	Description	Values	Units
1: Instance	Relay instance number.	*min:*0 *increment:*1	
2: Count	Cycle count.	*min:*1 *increment:*1	
3: Time	Cycle time.	*min:*0	s

4	Empty
5	Empty
6	Empty
7	Empty

MAV_CMD_DO_SET_SERVO (183

)

[Command] Set a servo to a desired PWM value.

Param (:Label)	Description	Values	Units
1: Instance	Servo instance number.	*min:*0 *increment:*1	
2: PWM	Pulse Width Modulation.	*min:*0 *increment:*1	us
3	Empty		
4	Empty		
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_DO_REPEAT_SERVO (184

)

[Command] Cycle a between its nominal setting and a desired PWM for a desired number of cycles with a desired period.

Param (:Label)	Description	Values	Units
1: Instance	Servo instance number.	*min:*0 *increment:*1	
2: PWM	Pulse Width Modulation.	*min:*0 *increment:*1	us
3: Count	Cycle count.	*min:*1 *increment:*1	
4: Time	Cycle time.	*min:*0	s
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_DO_FLIGHTTERMINATION (185

)

[Command] Terminate flight immediately. Flight termination immediately and irreversably terminates the current flight, returning the vehicle to ground. The vehicle will ignore RC or other input until it has been power-cycled. Termination may trigger safety measures, including: disabling motors and deployment of parachute on multicopters, and setting flight surfaces to initiate a landing pattern on fixed-wing). On multicopters without a parachute it may trigger a crash landing. Support for this command can be tested using the protocol bit: MAV_PROTOCOL_CAPABILITY_FLIGHT_TERMINATION. Support for this command can also be tested by sending the command with param1=0 (< 0.5); the ACK should be either MAV_RESULT_FAILED or MAV_RESULT_UNSUPPORTED.

Param (:Label)	Description	Values
1: Terminate	Flight termination activated if > 0.5. Otherwise not activated and ACK with MAV_RESULT_FAILED.	*min:*0 *max:*1 *increment:*1
2	Empty	
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_CHANGE_ALTITUDE (186

)

[Command] Change altitude set point.

Param (:Label)	Description	Values	Units
1: Altitude	Altitude.		m
2: Frame	Frame of new altitude. MAV_FRAME		
3	Empty		
4	Empty		
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_DO_SET_ACTUATOR (187

)

[Command] Sets actuators (e.g. servos) to a desired value. The actuator numbers are mapped to specific outputs (e.g. on any MAIN or AUX PWM or UAVCAN) using a flight-stack specific mechanism (i.e. a parameter).

Param (:Label)	Description	Values
1: Actuator 1	Actuator 1 value, scaled from [-1 to 1]. NaN to ignore.	*min:* -1 *max:*1
2: Actuator 2	Actuator 2 value, scaled from [-1 to 1]. NaN to ignore.	*min:* -1 *max:*1

3: Actuator 3	Actuator 3 value, scaled from [-1 to 1]. NaN to ignore.	*min:* -1 *max:*1
4: Actuator 4	Actuator 4 value, scaled from [-1 to 1]. NaN to ignore.	*min:* -1 *max:*1
5: Actuator 5	Actuator 5 value, scaled from [-1 to 1]. NaN to ignore.	*min:* -1 *max:*1
6: Actuator 6	Actuator 6 value, scaled from [-1 to 1]. NaN to ignore.	*min:* -1 *max:*1
7: Index	Index of actuator set (i.e if set to 1, Actuator 1 becomes Actuator 7)	*min:*0 *increment:*1

MAV_CMD_DO_LAND_START (189

)

[Command] Mission command to perform a landing. This is used as a marker in a mission to tell the autopilot where a sequence of mission items that represents a landing starts. It may also be sent via a COMMAND_LONG to trigger a landing, in which case the nearest (geographically) landing sequence in the mission will be used. The Latitude/Longitude is optional, and may be set to 0 if not needed. If specified then it will be used to help find the closest landing sequence.

Param (:Label)	Description
1	Empty
2	Empty
3	Empty
4	Empty
5: Latitude	Latitude
6: Longitude	Longitude
7	Empty

MAV_CMD_DO_RALLY_LAND (190

)

[Command] Mission command to perform a landing from a rally point.

Param (:Label)	Description	Units
1: Altitude	Break altitude	m
2: Speed	Landing speed	m/s
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_GO_AROUND (191

)

[Command] Mission command to safely abort an autonomous landing.

Param (:Label)	Description	Units
1: Altitude	Altitude	m
2	Empty	
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_REPOSITION (192

)

[Command] Reposition the vehicle to a specific WGS84 global position.

Param (:Label)	Description	Values	Units
1: Speed	Ground speed, less than 0 (-1) for default	*min:* -1	m/s
2: Bitmask	Bitmask of option flags.	MAV_DO_REPOSITION_FLAGS	
3: Radius	Loiter radius for planes. Positive values only, direction is controlled by Yaw value. A value of zero or NaN is ignored.		m
4: Yaw	Yaw heading. NaN to use the current system yaw heading mode (e.g. yaw towards next waypoint, yaw to home, etc.). For planes indicates loiter direction (0: clockwise, 1: counter clockwise)		deg
5: Latitude	Latitude		
6: Longitude	Longitude		
7:	Altitude		m

Altitude

MAV_CMD_DO_PAUSE_CONTINUE (193

)

[Command] If in a GPS controlled position mode, hold the current position or continue.

Param (:Label)	Description	Values
1: Continue	0: Pause current mission or reposition command, hold current position. 1: Continue mission. A VTOL capable vehicle should enter hover mode (multicopter and VTOL planes). A plane should loiter with the default loiter radius.	*min:*0 *max:*1 *increment:*1
2	Reserved	
3	Reserved	
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	

MAV_CMD_DO_SET_REVERSE (194

)

[Command] Set moving direction to forward or reverse.

Param (:Label)	Description	Values
1: Reverse	Direction (0=Forward, 1=Reverse)	*min:*0 *max:*1 *increment:*1
2	Empty	
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_SET_ROI_LOCATION (195

)

[Command] Sets the region of interest (ROI) to a location. This can then be used by the vehicle's control system to control the vehicle attitude and the attitude of various sensors such as cameras. This command can be sent to a gimbal manager but not to a gimbal device. A gimbal is not to react to this message.

Param (:Label)	Description	Units
1: Gimbal device ID	Component ID of gimbal device to address (or 1-6 for non-MAVLink gimbal), 0 for all gimbal device components. Send command multiple times for more than one gimbal (but not all gimbals).	
2	Empty	
3	Empty	
4	Empty	
5: Latitude	Latitude of ROI location	degE7
6: Longitude	Longitude of ROI location	degE7
7: Altitude	Altitude of ROI location	m

MAV_CMD_DO_SET_ROI_WPNEXT_OFFSET (196

)

[Command] Sets the region of interest (ROI) to be toward next waypoint, with optional pitch/roll/yaw offset. This can then be used by the vehicle's control system to control the vehicle attitude and the attitude of various sensors such as cameras. This command can be sent to a gimbal manager but not to a gimbal device. A gimbal device is not to react to this message.

Param (:Label)	Description
1: Gimbal device ID	Component ID of gimbal device to address (or 1-6 for non-MAVLink gimbal), 0 for all gimbal device components. Send command multiple times for more than one gimbal (but not all gimbals).
2	Empty
3	Empty
4	Empty
5: Pitch Offset	Pitch offset from next waypoint, positive pitching up
6: Roll Offset	Roll offset from next waypoint, positive rolling to the right
7: Yaw Offset	Yaw offset from next waypoint, positive yawing to the right

MAV_CMD_DO_SET_ROI_NONE (197

)

[Command] Cancels any previous ROI command returning the vehicle/sensors to default flight characteristics. This can then be used by the vehicle's control system to control the vehicle attitude and the attitude of various sensors such as cameras. This command can be sent to a gimbal manager but not to a gimbal device. A gimbal device is not to react to this message. After this command the gimbal manager should go back to manual input if available, and otherwise assume a neutral position.

Param (:Label)	Description

1: Gimbal device ID	Component ID of gimbal device to address (or 1-6 for non-MAVLink gimbal), 0 for all gimbal device components. Send command multiple times for more than one gimbal (but not all gimbals).
2	Empty
3	Empty
4	Empty
5	Empty
6	Empty
7	Empty

MAV_CMD_DO_SET_ROI_SYSID (198

)

[Command] Mount tracks system with specified system ID. Determination of target vehicle position may be done with GLOBAL_POSITION_INT or any other means. This command can be sent to a gimbal manager but not to a gimbal device. A gimbal device is not to react to this message.

Param (:Label)	Description	Values
1: System ID	System ID	*min:*1 *max:*255 *increment:*1
2: Gimbal device ID	Component ID of gimbal device to address (or 1-6 for non-MAVLink gimbal), 0 for all gimbal device components. Send command multiple times for more than one gimbal (but not all gimbals).	

MAV_CMD_DO_CONTROL_VIDEO (200

)

[Command] Control onboard camera system.

Param (:Label)	Description	Values	Units
1: ID	Camera ID (-1 for all)	*min:* -1 *increment:*1	
2: Transmission	Transmission: 0: disabled, 1: enabled compressed, 2: enabled raw	*min:*0 *max:*2 *increment:*1	
3: Interval	Transmission mode: 0: video stream, >0: single images every n seconds	*min:*0	s
4: Recording	Recording: 0: disabled, 1: enabled compressed, 2: enabled raw	*min:*0 *max:*2 *increment:*1	
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_DO_SET_ROI (201

)

DEPRECATED: Replaced by MAV_CMD_DO_SET_ROI_* (2018-01).

[Command] Sets the region of interest (ROI) for a sensor set or the vehicle itself. This can then be used by the vehicle's control system to control the vehicle attitude and the attitude of various sensors such as cameras.

Param (:Label)	Description	Values
1: ROI Mode	Region of interest mode.	MAV_ROI
2: WP Index	Waypoint index/ target ID (depends on param 1).	*min:*0 *increment:*1
3: ROI Index	Region of interest index. (allows a vehicle to manage multiple ROI's)	*min:*0 *increment:*1
4	Empty	
5	MAV_ROI_WPNEXT: pitch offset from next waypoint, MAV_ROI_LOCATION: latitude	
6	MAV_ROI_WPNEXT: roll offset from next waypoint, MAV_ROI_LOCATION: longitude	
7	MAV_ROI_WPNEXT: yaw offset from next waypoint, MAV_ROI_LOCATION: altitude	

MAV_CMD_DO_DIGICAM_CONFIGURE (202

)

[Command] Configure digital camera. This is a fallback message for systems that have not yet implemented PARAM_EXT_XXX messages and camera definition files (see https://mavlink.io/en/services/camera_def.html).

Param (:Label)	Description	Values	Units
1: Mode	Modes: P, TV, AV, M, Etc.	*min:*0 *increment:*1	
2: Shutter Speed	Shutter speed: Divisor number for one second.	*min:*0 *increment:*1	
3: Aperture	Aperture: F stop number.	*min:*0	
4: ISO	ISO number e.g. 80, 100, 200, Etc.	*min:*0 *increment:*1	
5: Exposure	Exposure type enumerator.		
6: Command Identity	Command Identity.		
7: Engine Cut-off	Main engine cut-off time before camera trigger. (0 means no cut-off)	*min:*0 *increment:*1	ds

MAV_CMD_DO_DIGICAM_CONTROL (203

)

[Command] Control digital camera. This is a fallback message for systems that have not yet implemented PARAM_EXT_XXX messages and camera definition files (see https://mavlink.io/en/services/camera_def.html).

Param (:Label)	Description
1: Session Control	Session control e.g. show/hide lens
2: Zoom Absolute	Zoom's absolute position
3: Zoom Relative	Zooming step value to offset zoom from the current position

4: Focus	Focus Locking, Unlocking or Re-locking
5: Shoot Command	Shooting Command
6: Command Identity	Command Identity
7: Shot ID	Test shot identifier. If set to 1, image will only be captured, but not counted towards internal frame count.

MAV_CMD_DO_MOUNT_CONFIGURE (204

)

DEPRECATED: Replaced by MAV_CMD_DO_GIMBAL_MANAGER_CONFIGURE (2020-01).

This message has been superseded by MAV_CMD_DO_GIMBAL_MANAGER_CONFIGURE. The message can still be used to communicate with legacy gimbals implementing it.

[Command] Mission command to configure a camera or antenna mount

Param (:Label)	Description	Values
1: Mode	Mount operation mode	MAV_MOUNT_MODE
2: Stabilize Roll	stabilize roll? (1 = yes, 0 = no)	*min:*0 *max:*1 *increment:*1
3: Stabilize Pitch	stabilize pitch? (1 = yes, 0 = no)	*min:*0 *max:*1 *increment:*1
4: Stabilize Yaw	stabilize yaw? (1 = yes, 0 = no)	*min:*0 *max:*1 *increment:*1
5: Roll Input Mode	roll input (0 = angle body frame, 1 = angular rate, 2 = angle absolute frame)	
6: Pitch Input Mode	pitch input (0 = angle body frame, 1 = angular rate, 2 = angle absolute frame)	
7: Yaw Input Mode	yaw input (0 = angle body frame, 1 = angular rate, 2 = angle absolute frame)	

MAV_CMD_DO_MOUNT_CONTROL (205

)

DEPRECATED: Replaced by MAV_CMD_DO_GIMBAL_MANAGER_PITCHYAW (2020-01).

This message is ambiguous and inconsistent. It has been superseded by MAV_CMD_DO_GIMBAL_MANAGER_PITCHYAW and MAV_CMD_DO_SET_ROI_*. The message can still be used to communicate with legacy gimbals implementing it.

[Command] Mission command to control a camera or antenna mount

Param (:Label)	Description	Values	Units
1: Pitch	pitch depending on mount mode (degrees or degrees/second depending on pitch input).		
2: Roll	roll depending on mount mode (degrees or degrees/second depending on roll input).		
3: Yaw	yaw depending on mount mode (degrees or degrees/second depending on yaw input).		
4: Altitude	altitude depending on mount mode.		m
5: Latitude	latitude, set if appropriate mount mode.		
6: Longitude	longitude, set if appropriate mount mode.		
7: Mode	Mount mode.	MAV_MOUNT_MODE	

MAV_CMD_DO_SET_CAM_TRIGG_DIST (206

)

[Command] Mission command to set camera trigger distance for this flight. The camera is triggered each time this distance is exceeded. This command can also be used to set the shutter integration time for the camera.

Param (:Label)	Description	Values	Units
1: Distance	Camera trigger distance. 0 to stop triggering.	*min:*0	m
2: Shutter	Camera shutter integration time. -1 or 0 to ignore	*min:*	
-1			
*increment:*1	ms		
3: Trigger	Trigger camera once immediately. (0 = no trigger, 1 = trigger)	*min:*0 *max:*1 *increment:*1	
4	Empty		
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_DO_FENCE_ENABLE (207

)

[Command] Mission command to enable the geofence

Param (:Label)	Description	Values
1: Enable	enable? (0=disable, 1=enable, 2=disable_floor_only)	*min:*0 *max:*2 *increment:*1
2	Empty	
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_PARACHUTE (208

)

[Command] Mission item/command to release a parachute or enable/disable auto release.

Param (:Label)	Description	Values
1: Action	Action	PARACHUTE_ACTION
2	Empty	

3	Empty
4	Empty
5	Empty
6	Empty
7	Empty

MAV_CMD_DO_MOTOR_TEST ([209

)

[Command] Command to perform motor test.

Param (:Label)	Description	Values	Units
1: Instance	Motor instance number (from 1 to max number of motors on the vehicle).	*min:*1 *increment:*1	
2: Throttle Type	Throttle type (whether the Throttle Value in param3 is a percentage, PWM value, etc.)	MOTOR_TEST_THROTTLE_TYPE	
3: Throttle	Throttle value.		
4: Timeout	Timeout between tests that are run in sequence.	*min:*0	s
5: Motor Count	Motor count. Number of motors to test in sequence: 0/1=one motor, 2= two motors, etc. The Timeout (param4) is used between tests.	*min:*0 *increment:*1	
6: Test Order	Motor test order.	MOTOR_TEST_ORDER	
7	Empty		

MAV_CMD_DO_INVERTED_FLIGHT ([210

)

[Command] Change to/from inverted flight.

Param (:Label)	Description	Values
1: Inverted	Inverted flight. (0=normal, 1=inverted)	*min:*0 *max:*1 *increment:*1
2	Empty	
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_GRIPPER ([211

)

[Command] Mission command to operate a gripper.

Param (:Label)	Description	Values
1: Instance	Gripper instance number.	*min:*1 *increment:*1
2: Action	Gripper action to perform.	GRIPPER_ACTIONS
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_AUTOTUNE_ENABLE ([212

)

[Command] Enable/disable autotune.

Param (:Label)	Description	Values
1: Enable	Enable (1: enable, 0:disable).	*min:*0 *max:*1 *increment:*1
2: Axis	Specify which axis are autotuned. 0 indicates autopilot default settings.	AUTOTUNE_AXIS
3	Empty.	
4	Empty.	
5	Empty.	
6	Empty.	
7	Empty.	

MAV_CMD_NAV_SET_YAW_SPEED ([213

)

[Command] Sets a desired vehicle turn angle and speed change.

Param (:Label)	Description	Values	Units
1: Yaw	Yaw angle to adjust steering by.		deg
2: Speed	Speed.		m/s
3: Angle	Final angle. (0=absolute, 1=relative)	*min:*0 *max:*1 *increment:*1	
4	Empty		
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_DO_SET_CAM_TRIGG_INTERVAL ([214

)

[Command] Mission command to set camera trigger interval for this flight. If triggering is enabled, the camera is triggered each time this interval expires. This command can also be used to set the shutter integration time for the camera.

Param (:Label)	Description	Values	Units
1: Trigger Cycle	Camera trigger cycle time. -1 or 0 to ignore.	*min:* -1 *increment:*1	ms
2: Shutter Integration	Camera shutter integration time. Should be less than trigger cycle time. -1 or 0 to ignore.	*min:* -1 *increment:*1	ms
3	Empty		
4	Empty		
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_DO_MOUNT_CONTROL_QUAT (220

)

DEPRECATED: Replaced by MAV_CMD_DO_GIMBAL_MANAGER_PITCHYAW (2020-01).

[Command] Mission command to control a camera or antenna mount, using a quaternion as reference.

Param (:Label)	Description
1: Q1	quaternion param q1, w (1 in null-rotation)
2: Q2	quaternion param q2, x (0 in null-rotation)
3: Q3	quaternion param q3, y (0 in null-rotation)
4: Q4	quaternion param q4, z (0 in null-rotation)
5	Empty
6	Empty
7	Empty

MAV_CMD_DO_GUIDED_MASTER (221

)

[Command] set id of master controller

Param (:Label)	Description	Values
1: System ID	System ID	*min:*0 *max:*255 *increment:*1
2: Component ID	Component ID	*min:*0 *max:*255 *increment:*1
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_GUIDED_LIMITS (222

)

[Command] Set limits for external control

Param (:Label)	Description	Values	Units
1: Timeout	Timeout - maximum time that external controller will be allowed to control vehicle. 0 means no timeout.	*min:*0	s
2: Min Altitude	Altitude (MSL) min - if vehicle moves below this alt, the command will be aborted and the mission will continue. 0 means no lower altitude limit.		m
3: Max Altitude	Altitude (MSL) max - if vehicle moves above this alt, the command will be aborted and the mission will continue. 0 means no upper altitude limit.		m
4: Horiz. Move Limit	Horizontal move limit - if vehicle moves more than this distance from its location at the moment the command was executed, the command will be aborted and the mission will continue. 0 means no horizontal move limit.	*min:*0	m
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_DO_ENGINE_CONTROL (223

)

[Command] Control vehicle engine. This is interpreted by the vehicles engine controller to change the target engine state. It is intended for vehicles with internal combustion engines

Param (:Label)	Description	Values	Units
1: Start Engine	0: Stop engine, 1:Start Engine	*min:*0 *max:*1 *increment:*1	
2: Cold Start	0: Warm start, 1:Cold start. Controls use of choke where applicable	*min:*0 *max:*1 *increment:*1	
3: Height	Height delay. This is for commanding engine start only after the vehicle has gained the specified height. Used in VTOL vehicles during takeoff to start engine after the aircraft is off the ground. Zero for no	*min:*0	m

Delay	delay.
4	Empty
5	Empty
5	Empty
6	Empty
7	Empty

MAV_CMD_DO_SET_MISSION_CURRENT (224

)

[Command] Set the mission item with sequence number seq as current item. This means that the MAV will continue to this mission item on the shortest path (not following the mission items in-between).

Param (:Label)	Description	Values
1: Number	Mission sequence value to set	*min:*0 *increment:*1
2	Empty	
3	Empty	
4	Empty	
5	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_DO_LAST (240

)

[Command] NOP - This command is only used to mark the upper limit of the DO commands in the enumeration

Param (:Label)	Description
1	Empty
2	Empty
3	Empty
4	Empty
5	Empty
6	Empty
7	Empty

MAV_CMD_PREFLIGHT_CALIBRATION (241

)

[Command] Trigger calibration. This command will be only accepted if in pre-flight mode. Except for Temperature Calibration, only one sensor should be set in a single message and all others should be zero.

Param (:Label)	Description	Values
1: Gyro Temperature	1: gyro calibration, 3: gyro temperature calibration	*min:*0 *max:*3 *increment:*1
2: Magnetometer	1: magnetometer calibration	*min:*0 *max:*1 *increment:*1
3: Ground Pressure	1: ground pressure calibration	*min:*0 *max:*1 *increment:*1
4: Remote Control	1: radio RC calibration, 2: RC trim calibration	*min:*0 *max:*1 *increment:*1
5: Accelerometer	1: accelerometer calibration, 2: board level calibration, 3: accelerometer temperature calibration, 4: simple accelerometer calibration	*min:*0 *max:*4 *increment:*1
6: Compmot or Airspeed	1: APM: compass/motor interference calibration (PX4: airspeed calibration, deprecated), 2: airspeed calibration	*min:*0 *max:*2 *increment:*1
7: ESC or Baro	1: ESC calibration, 3: barometer temperature calibration	*min:*0 *max:*3 *increment:*1

MAV_CMD_PREFLIGHT_SET_SENSOR_OFFSETS (242

)

[Command] Set sensor offsets. This command will be only accepted if in pre-flight mode.

Param (:Label)	Description	Values
1: Sensor Type	Sensor to adjust the offsets for: 0: gyros, 1: accelerometer, 2: magnetometer, 3: barometer, 4: optical flow, 5: second magnetometer, 6: third magnetometer	*min:*0 *max:*6 *increment:*1
2: X Offset	X axis offset (or generic dimension 1), in the sensor's raw units	
3: Y Offset	Y axis offset (or generic dimension 2), in the sensor's raw units	
4: Z Offset	Z axis offset (or generic dimension 3), in the sensor's raw units	
5: 4th Dimension	Generic dimension 4, in the sensor's raw units	
6: 5th Dimension	Generic dimension 5, in the sensor's raw units	
7: 6th Dimension	Generic dimension 6, in the sensor's raw units	

MAV_CMD_PREFLIGHT_UAVCAN (243

)

[Command] Trigger UAVCAN configuration (actuator ID assignment and direction mapping). Note that this maps to the legacy UAVCAN

v0 function UAVCAN_ENUMERATE, which is intended to be executed just once during initial vehicle configuration (it is not a normal pre-flight command and has been poorly named).

Param (:Label)	Description
1: Actuator ID	1: Trigger actuator ID assignment and direction mapping. 0: Cancel command.
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved

MAV_CMD_PREFLIGHT_STORAGE (245

)

[Command] Request storage of different parameter values and logs. This command will be only accepted if in pre-flight mode.

Param (:Label)	Description	Values	Units
1: Parameter Storage	Action to perform on the persistent parameter storage	PREFLIGHT_STORAGE_PARAMETER_ACTION	
2: Mission Storage	Action to perform on the persistent mission storage	PREFLIGHT_STORAGE_MISSION_ACTION	
3: Logging Rate	Onboard logging: 0: Ignore, 1: Start default rate logging, -1: Stop logging, > 1: logging rate (e.g. set to 1000 for 1000 Hz logging)	*min:* -1 *increment:*1	Hz
4	Reserved		
5	Empty		
6	Empty		
7	Empty		

MAV_CMD_PREFLIGHT_REBOOT_SHUTDOWN (246

)

[Command] Request the reboot or shutdown of system components.

Param (:Label)	Description	Values
1: Autopilot	0: Do nothing for autopilot, 1: Reboot autopilot, 2: Shutdown autopilot, 3: Reboot autopilot and keep it in the bootloader until upgraded.	*min:*0 *max:*3 *increment:*1
2: Companion	0: Do nothing for onboard computer, 1: Reboot onboard computer, 2: Shutdown onboard computer, 3: Reboot onboard computer and keep it in the bootloader until upgraded.	*min:*0 *max:*3 *increment:*1
3: Component action	0: Do nothing for component, 1: Reboot component, 2: Shutdown component, 3: Reboot component and keep it in the bootloader until upgraded	*min:*0 *max:*3 *increment:*1
4: Component ID	MAVLink Component ID targeted in param3 (0 for all components).	*min:*0 *max:*255 *increment:*1
5	Reserved (set to 0)	
6	Reserved (set to 0)	
7	WIP: ID (e.g. camera ID -1 for all IDs)	

MAV_CMD_OVERRIDE_GOTO (252

)

[Command] Override current mission with command to pause mission, pause mission and move to position, continue/resume mission. When param 1 indicates that the mission is paused (MAV_GOTO_DO_HOLD), param 2 defines whether it holds in place or moves to another position.

Param (:Label)	Description	Values	Units
1: Continue	MAV_GOTO_DO_HOLD: pause mission and either hold or move to specified position (depending on param2), MAV_GOTO_DO_CONTINUE: resume mission.	MAV_GOTO	
2: Position	MAV_GOTO_HOLD_AT_CURRENT_POSITION: hold at current position, MAV_GOTO_HOLD_AT_SPECIFIED_POSITION: hold at specified position.	MAV_GOTO	
3: Frame	Coordinate frame of hold point.	MAV_FRAME	
4: Yaw	Desired yaw angle.		deg
5: Latitude/X	Latitude/X position.		
6: Longitude/Y	Longitude/Y position.		
7: Altitude/Z	Altitude/Z position.		

MAV_CMD_OBLIQUE_SURVEY (260

)

[Command] Mission command to set a Camera Auto Mount Pivoting Oblique Survey (Replaces CAM_TRIGG_DIST for this purpose). The camera is triggered each time this distance is exceeded, then the mount moves to the next position. Params 4~6 set-up the angle limits and number of positions for oblique survey, where mount-enabled vehicles automatically roll the camera between shots to emulate an oblique camera setup (providing an increased HFOV). This command can also be used to set the shutter integration time for the camera.

Param

(:Label)	Description	Values	Units
1: Distance	Camera trigger distance. 0 to stop triggering.	*min:*0	m
2: Shutter	Camera shutter integration time. 0 to ignore	*min:*0 *increment:*1	ms
3: Min Interval	The minimum interval in which the camera is capable of taking subsequent pictures repeatedly. 0 to ignore.	*min:*0 *max:*10000 *increment:*1	ms
4: Positions	Total number of roll positions at which the camera will capture photos (images captures spread evenly across the limits defined by param5).	*min:*2 *increment:*1	
5: Roll Angle	Angle limits that the camera can be rolled to left and right of center.	*min:*0	deg
6: Pitch Angle	Fixed pitch angle that the camera will hold in oblique mode if the mount is actuated in the pitch axis.	*min:* -180 *max:*180	deg
7	Empty		

MAV_CMD_MISSION_START (300

)

[Command] start running a mission

Param (:Label)	Description	Values
1: First Item	first_item: the first mission item to run	*min:*0 *increment:*1
2: Last Item	last_item: the last mission item to run (after this item is run, the mission ends)	*min:*0 *increment:*1

MAV_CMD_ACTUATOR_TEST (310

)

[Command] Actuator testing command. This is similar to MAV_CMD_DO_MOTOR_TEST but operates on the level of output functions, i.e. it is possible to test Motor1 independent from which output it is configured on. Autopilots typically refuse this command while armed.

Param (:Label)	Description	Values	Units
1: Value	Output value: 1 means maximum positive output, 0 to center servos or minimum motor thrust (expected to spin), -1 for maximum negative (if not supported by the motors, i.e. motor is not reversible, smaller than 0 maps to NaN). And NaN maps to disarmed (stop the motors).	*min: -1 *max:*1	
2: Timeout	Timeout after which the test command expires and the output is restored to the previous value. A timeout has to be set for safety reasons. A timeout of 0 means to restore the previous value immediately.	*min:*0 *max:*3	s
3	Reserved (set to 0)		
4	Reserved (set to 0)		
5: Output Function	Actuator Output function	ACTUATOR_OUTPUT_FUNCTION	
6	Reserved (set to 0)		
7	Reserved (set to 0)		

MAV_CMD_CONFIGURE_ACTUATOR (311

)

[Command] Actuator configuration command.

Param (:Label)	Description	Values
1: Configuration	Actuator configuration action	ACTUATOR_CONFIGURATION
2	Reserved (set to 0)	
3	Reserved (set to 0)	
4	Reserved (set to 0)	
5: Output Function	Actuator Output function	ACTUATOR_OUTPUT_FUNCTION
6	Reserved (set to 0)	
7	Reserved (set to 0)	

MAV_CMD_COMPONENT_ARM_DISARM (400

)

[Command] Arms / Disarms a component

Param (:Label)	Description	Values
1: Arm	0: disarm, 1: arm	*min:*0 *max:*1 *increment:*1
2: Force	0: arm-disarm unless prevented by safety checks (i.e. when landed), 21196: force arming/disarming (e.g. allow arming to override preflight checks and disarming in flight)	*min:*0 *max:*21196 *increment:*21196

MAV_CMD_RUN_PREARM_CHECKS (401

)

[Command] Instructs a target system to run pre-arm checks. This allows preflight checks to be run on demand, which may be useful on systems that normally run them at low rate, or which do not trigger checks when the armable state might have changed. This command should return MAV_RESULT_ACCEPTED if it will run the checks. The results of the checks are usually then reported in SYS_STATUS

messages (this is system-specific). The command should return MAV_RESULT_TEMPORARILY_REJECTED if the system is already armed.

Param (:Label)	Description

MAV_CMD_ILLUMINATOR_ON_OFF (405

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Command] Turns illuminators ON/OFF. An illuminator is a light source that is used for lighting up dark areas external to the sytstem: e.g. a torch or searchlight (as opposed to a light source for illuminating the system itself, e.g. an indicator light).

Param (:Label)	Description	Values
1: Enable	0: Illuminators OFF, 1: Illuminators ON	*min:*0 *max:*1 *increment:*1

MAV_CMD_GET_HOME_POSITION (410

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2022-04).

[Command] Request the home position from the vehicle. The vehicle will ACK the command and then emit the HOME_POSITION message.

Param (:Label)	Description
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved

MAV_CMD_INJECT_FAILURE (420

)

[Command] Inject artificial failure for testing purposes. Note that autopilots should implement an additional protection before accepting this command such as a specific param setting.

Param (:Label)	Description	Values
1: Failure unit	The unit which is affected by the failure.	FAILURE_UNIT
2: Failure type	The type how the failure manifests itself.	FAILURE_TYPE
3: Instance	Instance affected by failure (0 to signal all).	

MAV_CMD_START_RX_PAIR (500

)

[Command] Starts receiver pairing.

Param (:Label)	Description	Values
1: Spektrum	0:Spektrum.	
2: RC Type	RC type.	RC_TYPE

MAV_CMD_GET_MESSAGE_INTERVAL (510

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2022-04).

[Command] Request the interval between messages for a particular MAVLink message ID. The receiver should ACK the command and then emit its response in a MESSAGE_INTERVAL message.

Param (:Label)	Description	Values
1: Message ID	The MAVLink message ID	*min:*0 *max:*16777215 *increment:*1

MAV_CMD_SET_MESSAGE_INTERVAL (511

)

[Command] Set the interval between messages for a particular MAVLink message ID. This interface replaces REQUEST_DATA_STREAM.

Param (:Label)	Description	Values	Units
1: Message ID	The MAVLink message ID	*min:*0 *max:*16777215 *increment:*1	
2: Interval	The interval between two messages. Set to -1 to disable and 0 to request default rate.	*min:* -1 *increment:*1	us
7: Response Target	Target address of message stream (if message has target address fields). 0: Flight-stack default (recommended), 1: address of requestor, 2: broadcast.	*min:*0 *max:*2 *increment:*1	

MAV_CMD_REQUEST_MESSAGE (512

)

[Command] Request the target system(s) emit a single instance of a specified message (i.e. a "one-shot" version of MAV_CMD_SET_MESSAGE_INTERVAL).

Param (:Label)	Description	Values
1: Message ID	The MAVLink message ID of the requested message.	*min:*0 *max:*16777215 *increment:*1
2: Req Param 1	Use for index ID, if required. Otherwise, the use of this parameter (if any) must be defined in the requested message. By default assumed not used (0).	
3: Req Param 2	The use of this parameter (if any), must be defined in the requested message. By default assumed not used (0).	
4: Req Param 3	The use of this parameter (if any), must be defined in the requested message. By default assumed not used (0).	
5: Req Param 4	The use of this parameter (if any), must be defined in the requested message. By default assumed not used (0).	
6: Req Param 5	The use of this parameter (if any), must be defined in the requested message. By default assumed not used (0).	
7: Response Target	Target address for requested message (if message has target address fields). 0: Flight-stack default, 1: address of requestor, 2: broadcast.	*min:*0 *max:*2 *increment:*1

MAV_CMD_REQUEST_PROTOCOL_VERSION (519

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Request MAVLink protocol version compatibility. All receivers should ACK the command and then emit their capabilities in an PROTOCOL_VERSION message

Param (:Label)	Description	Values
1: Protocol	1: Request supported protocol versions by all nodes on the network	*min:*0 *max:*1 *increment:*1
2	Reserved (all remaining params)	

MAV_CMD_REQUEST_AUTOPILOT_CAPABILITIES (520

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Request autopilot capabilities. The receiver should ACK the command and then emit its capabilities in an AUTOPILOT_VERSION message

Param (:Label)	Description	Values
1: Version	1: Request autopilot version	*min:*0 *max:*1 *increment:*1
2	Reserved (all remaining params)	

MAV_CMD_REQUEST_CAMERA_INFORMATION (521

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Request camera information (CAMERA_INFORMATION).

Param (:Label)	Description	Values
1: Capabilities	0: No action 1: Request camera capabilities	*min:*0 *max:*1 *increment:*1
2	Reserved (all remaining params)	

MAV_CMD_REQUEST_CAMERA_SETTINGS (522

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Request camera settings (CAMERA_SETTINGS).

Param (:Label)	Description	Values
1: Settings	0: No Action 1: Request camera settings	*min:*0 *max:*1 *increment:*1
2	Reserved (all remaining params)	

MAV_CMD_REQUEST_STORAGE_INFORMATION (525

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Request storage information (STORAGE_INFORMATION). Use the command's target_component to target a specific component's storage.

Param (:Label)	Description	Values
1: Storage ID	Storage ID (0 for all, 1 for first, 2 for second, etc.)	*min:*0 *increment:*1
2: Information	0: No Action 1: Request storage information	*min:*0 *max:*1 *increment:*1
3	Reserved (all remaining params)	

MAV_CMD_STORAGE_FORMAT (526

)

[Command] Format a storage medium. Once format is complete, a STORAGE_INFORMATION message is sent. Use the command's target_component to target a specific component's storage.

Param (:Label)	Description	Values
1: Storage ID	Storage ID (1 for first, 2 for second, etc.)	*min:*0 *increment:*1
2: Format	Format storage (and reset image log). 0: No action 1: Format storage	*min:*0 *max:*1 *increment:*1
3: Reset Image Log	Reset Image Log (without formatting storage medium). This will reset CAMERA_CAPTURE_STATUS.image_count and CAMERA_IMAGE_CAPTURED.image_index. 0: No action 1: Reset Image Log	*min:*0 *max:*1 *increment:*1
4	Reserved (all remaining params)	

MAV_CMD_REQUEST_CAMERA_CAPTURE_STATUS (527

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Request camera capture status (CAMERA_CAPTURE_STATUS)

Param (:Label)	Description	Values
1: Capture Status	0: No Action 1: Request camera capture status	*min:*0 *max:*1 *increment:*1
2	Reserved (all remaining params)	

MAV_CMD_REQUEST_FLIGHT_INFORMATION (528

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Request flight information (FLIGHT_INFORMATION)

Param (:Label)	Description	Values
1: Flight Information	1: Request flight information	*min:*0 *max:*1 *increment:*1
2	Reserved (all remaining params)	

MAV_CMD_RESET_CAMERA_SETTINGS (529

)

[Command] Reset all camera settings to Factory Default

Param (:Label)	Description	Values
1: Reset	0: No Action 1: Reset all settings	*min:*0 *max:*1 *increment:*1
2	Reserved (all remaining params)	

MAV_CMD_SET_CAMERA_MODE (530

)

[Command] Set camera running mode. Use NaN for reserved values. GCS will send a MAV_CMD_REQUEST_VIDEO_STREAM_STATUS command after a mode change if the camera supports video streaming.

Param (:Label)	Description	Values
1	Reserved (Set to 0)	
2: Camera Mode	Camera mode	CAMERA_MODE
3	Reserved (set to NaN)	
4	Reserved (set to NaN)	
7	Reserved (set to NaN)	

MAV_CMD_SET_CAMERA_ZOOM (531

)

[Command] Set camera zoom. Camera must respond with a CAMERA_SETTINGS message (on success).

Param (:Label)	Description	Values
1: Zoom Type	Zoom type	CAMERA_ZOOM_TYPE
2: Zoom Value	Zoom value. The range of valid values depend on the zoom type.	
3	Reserved (set to NaN)	
4	Reserved (set to NaN)	
7	Reserved (set to NaN)	

MAV_CMD_SET_CAMERA_FOCUS (532

)

[Command] Set camera focus. Camera must respond with a CAMERA_SETTINGS message (on success).

Param (:Label)	Description	Values
1: Focus Type	Focus type	SET_FOCUS_TYPE
2: Focus Value	Focus value	
3	Reserved (set to NaN)	
4	Reserved (set to NaN)	
7	Reserved (set to NaN)	

MAV_CMD_SET_STORAGE_USAGE (533

)

[Command] Set that a particular storage is the preferred location for saving photos, videos, and/or other media (e.g. to set that an SD card is used for storing videos). There can only be one preferred save location for each particular media type: setting a media usage flag will clear/reset that same flag if set on any other storage. If no flag is set the system should use its default storage. A target system can choose to always use default storage, in which case it should ACK the command with MAV_RESULT_UNSUPPORTED. A target system can choose to not allow a particular storage to be set as preferred storage, in which case it should ACK the command with MAV_RESULT_DENIED.

Param (:Label)	Description	Values
1: Storage ID	Storage ID (1 for first, 2 for second, etc.)	*min:*0 *increment:*1
2: Usage	Usage flags	STORAGE_USAGE_FLAG

MAV_CMD_JUMP_TAG (600

)

[Command] Tagged jump target. Can be jumped to with MAV_CMD_DO_JUMP_TAG.

Param (:Label)	Description	Values
1: Tag	Tag.	*min:*0 *increment:*1

MAV_CMD_DO_JUMP_TAG (601

)

[Command] Jump to the matching tag in the mission list. Repeat this action for the specified number of times. A mission should contain a single matching tag for each jump. If this is not the case then a jump to a missing tag should complete the mission, and a jump where there are multiple matching tags should always select the one with the lowest mission sequence number.

Param (:Label)	Description	Values
1: Tag	Target tag to jump to.	*min:*0 *increment:*1
2: Repeat	Repeat count.	*min:*0 *increment:*1

MAV_CMD_DO_GIMBAL_MANAGER_PITCHYAW (1000

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Command] High level setpoint to be sent to a gimbal manager to set a gimbal attitude. It is possible to set combinations of the values below. E.g. an angle as well as a desired angular rate can be used to get to this angle at a certain angular rate, or an angular rate only will result in continuous turning. NaN is to be used to signal unset. Note: a gimbal is never to react to this command but only the gimbal manager.

Param (:Label)	Description	Values	Units
1: Pitch angle	Pitch angle (positive to pitch up, relative to vehicle for FOLLOW mode, relative to world horizon for LOCK mode).	*min:* -180 *max:*180	deg
2: Yaw angle	Yaw angle (positive to yaw to the right, relative to vehicle for FOLLOW mode, absolute to North for LOCK mode).	*min:* -180 *max:*180	deg
3: Pitch rate	Pitch rate (positive to pitch up).		deg/s
4: Yaw rate	Yaw rate (positive to yaw to the right).		deg/s
5: Gimbal manager flags	Gimbal manager flags to use.	GIMBAL_MANAGER_FLAGS	
7: Gimbal device ID	Component ID of gimbal device to address (or 1-6 for non-MAVLink gimbal), 0 for all gimbal device components. Send command multiple times for more than one gimbal (but not all gimbals).		

MAV_CMD_DO_GIMBAL_MANAGER_CONFIGURE (1001

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Command] Gimbal configuration to set which sysid/compid is in primary and secondary control.

Param (:Label)	Description
1: sysid primary control	Sysid for primary control (0: no one in control, -1: leave unchanged, -2: set itself in control (for missions where the own sysid is still unknown), -3: remove control if currently in control).
2: compid primary control	Compid for primary control (0: no one in control, -1: leave unchanged, -2: set itself in control (for missions where the own sysid is still unknown), -3: remove control if currently in control).
3: sysid secondary control	Sysid for secondary control (0: no one in control, -1: leave unchanged, -2: set itself in control (for missions where the own sysid is still unknown), -3: remove control if currently in control).
4: compid secondary control	Compid for secondary control (0: no one in control, -1: leave unchanged, -2: set itself in control (for missions where the own sysid is still unknown), -3: remove control if currently in control).
7: Gimbal device ID	Component ID of gimbal device to address (or 1-6 for non-MAVLink gimbal), 0 for all gimbal device components. Send command multiple times for more than one gimbal (but not all gimbals).

MAV_CMD_IMAGE_START_CAPTURE (2000

)

[Command] Start image capture sequence. Sends CAMERA_IMAGE_CAPTURED after each capture. Use NaN for reserved values.

Param (:Label)	Description	Values	Units
1	Reserved (Set to 0)		
2: Interval	Desired elapsed time between two consecutive pictures (in seconds). Minimum values depend on hardware (typically greater than 2 seconds).	*min:*0	s
3: Total Images	Total number of images to capture. 0 to capture forever/until MAV_CMD_IMAGE_STOP_CAPTURE.	*min:*0 *increment:*1	
4: Sequence Number	Capture sequence number starting from 1. This is only valid for single-capture (param3 == 1), otherwise set to 0. Increment the capture ID for each capture command to prevent double captures when a command is re-transmitted.	*min:*1 *increment:*1	
5	Reserved (set to NaN)		
6	Reserved (set to NaN)		
7	Reserved (set to NaN)		

MAV_CMD_IMAGE_STOP_CAPTURE (2001

)

[Command] Stop image capture sequence Use NaN for reserved values.

Param (:Label)	Description
1	Reserved (Set to 0)
2	Reserved (set to NaN)
3	Reserved (set to NaN)
4	Reserved (set to NaN)
7	Reserved (set to NaN)

MAV_CMD_REQUEST_CAMERA_IMAGE_CAPTURE (2002

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Re-request a CAMERA_IMAGE_CAPTURED message.

Param (:Label)	Description	Values
1: Number	Sequence number for missing CAMERA_IMAGE_CAPTURED message	*min:*0 *increment:*1
2	Reserved (set to NaN)	
3	Reserved (set to NaN)	
4	Reserved (set to NaN)	
7	Reserved (set to NaN)	

MAV_CMD_DO_TRIGGER_CONTROL (2003

)

[Command] Enable or disable on-board camera triggering system.

Param (:Label)	Description	Values
1: Enable	Trigger enable/disable (0 for disable, 1 for start), -1 to ignore	*min:* -1 *max:*1 *increment:*1
2: Reset	1 to reset the trigger sequence, -1 or 0 to ignore	*min:* -1 *max:*1 *increment:*1
3: Pause	1 to pause triggering, but without switching the camera off or retracting it. -1 to ignore	*min:* -1 *max:*1 *increment:*2

MAV_CMD_CAMERA_TRACK_POINT (2004

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Command] If the camera supports point visual tracking (CAMERA_CAP_FLAGS_HAS_TRACKING_POINT is set), this command allows to initiate the tracking.

Param (:Label)	Description	Values
1: Point x	Point to track x value (normalized 0..1, 0 is left, 1 is right).	*min:*0 *max:*1
2: Point y	Point to track y value (normalized 0..1, 0 is top, 1 is bottom).	*min:*0 *max:*1
3: Radius	Point radius (normalized 0..1, 0 is image left, 1 is image right).	*min:*0 *max:*1

MAV_CMD_CAMERA_TRACK_RECTANGLE (2005

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Command] If the camera supports rectangle visual tracking (CAMERA_CAP_FLAGS_HAS_TRACKING_RECTANGLE is set), this command allows to initiate the tracking.

Param (:Label)	Description	Values
1: Top left corner x	Top left corner of rectangle x value (normalized 0..1, 0 is left, 1 is right).	*min:*0 *max:*1
2: Top left corner y	Top left corner of rectangle y value (normalized 0..1, 0 is top, 1 is bottom).	*min:*0 *max:*1
3: Bottom right corner x	Bottom right corner of rectangle x value (normalized 0..1, 0 is left, 1 is right).	*min:*0 *max:*1
4: Bottom right corner y	Bottom right corner of rectangle y value (normalized 0..1, 0 is top, 1 is bottom).	*min:*0 *max:*1

MAV_CMD_CAMERA_STOP_TRACKING (2010

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Command] Stops ongoing tracking.

Param (:Label)	Description

MAV_CMD_VIDEO_START_CAPTURE (2500

)

[Command] Starts video capture (recording).

Param (:Label)	Description	Values	Units
1: Stream ID	Video Stream ID (0 for all streams)	*min:*0 *increment:*1	
2: Status Frequency	Frequency CAMERA_CAPTURE_STATUS messages should be sent while recording (0 for no messages, otherwise frequency)	*min:*0	Hz
3	Reserved (set to NaN)		
4	Reserved (set to NaN)		
5	Reserved (set to NaN)		
6	Reserved (set to NaN)		
7	Reserved (set to NaN)		

MAV_CMD_VIDEO_STOP_CAPTURE (2501

)

[Command] Stop the current video capture (recording).

Param (:Label)	Description	Values
1: Stream ID	Video Stream ID (0 for all streams)	*min:*0 *increment:*1
2	Reserved (set to NaN)	
3	Reserved (set to NaN)	
4	Reserved (set to NaN)	
5	Reserved (set to NaN)	
6	Reserved (set to NaN)	
7	Reserved (set to NaN)	

MAV_CMD_VIDEO_START_STREAMING (2502

)

[Command] Start video streaming

Param (:Label)	Description	Values
1: Stream ID	Video Stream ID (0 for all streams, 1 for first, 2 for second, etc.)	*min:*0 *increment:*1

MAV_CMD_VIDEO_STOP_STREAMING (2503

)

[Command] Stop the given video stream

Param (:Label)	Description	Values
1: Stream ID	Video Stream ID (0 for all streams, 1 for first, 2 for second, etc.)	*min:*0 *increment:*1

MAV_CMD_REQUEST_VIDEO_STREAM_INFORMATION (2504

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Request video stream information (VIDEO_STREAM_INFORMATION)

Param (:Label)	Description	Values
1: Stream ID	Video Stream ID (0 for all streams, 1 for first, 2 for second, etc.)	*min:*0 *increment:*1

MAV_CMD_REQUEST_VIDEO_STREAM_STATUS (2505

)

DEPRECATED: Replaced by MAV_CMD_REQUEST_MESSAGE (2019-08).

[Command] Request video stream status (VIDEO_STREAM_STATUS)

Param (:Label)	Description	Values
1: Stream ID	Video Stream ID (0 for all streams, 1 for first, 2 for second, etc.)	*min:*0 *increment:*1

MAV_CMD_LOGGING_START (2510

)

[Command] Request to start streaming logging data over MAVLink (see also LOGGING_DATA message)

Param (:Label)	Description	Values

1: Format	Format: 0: ULog	*min:*0 *increment:*1
2	Reserved (set to 0)	
3	Reserved (set to 0)	
4	Reserved (set to 0)	
5	Reserved (set to 0)	
6	Reserved (set to 0)	
7	Reserved (set to 0)	

MAV_CMD_LOGGING_STOP (2511

)

[Command] Request to stop streaming log data over MAVLink

Param (:Label)	Description
1	Reserved (set to 0)
2	Reserved (set to 0)
3	Reserved (set to 0)
4	Reserved (set to 0)
5	Reserved (set to 0)
6	Reserved (set to 0)
7	Reserved (set to 0)

MAV_CMD_AIRFRAME_CONFIGURATION (2520

)

[Command]

Param (:Label)	Description	Values
1: Landing Gear ID	Landing gear ID (default: 0, -1 for all)	*min:* -1 *increment:*1
2: Landing Gear Position	Landing gear position (Down: 0, Up: 1, NaN for no change)	
3	Reserved (set to NaN)	
4	Reserved (set to NaN)	
5	Reserved (set to NaN)	
6	Reserved (set to NaN)	
7	Reserved (set to NaN)	

MAV_CMD_CONTROL_HIGH_LATENCY (2600

)

[Command] Request to start/stop transmitting over the high latency telemetry

Param (:Label)	Description	Values
1: Enable	Control transmission over high latency telemetry (0: stop, 1: start)	*min:*0 *max:*1 *increment:*1
2	Empty	
3	Empty	
4	Empty	
5	Empty	
6	Empty	
7	Empty	

MAV_CMD_PANORAMA_CREATE (2800

)

[Command] Create a panorama at the current position

Param (:Label)	Description	Units
1: Horizontal Angle	Viewing angle horizontal of the panorama (+- 0.5 the total angle)	deg
2: Vertical Angle	Viewing angle vertical of panorama.	deg
3: Horizontal Speed	Speed of the horizontal rotation.	deg/s
4: Vertical Speed	Speed of the vertical rotation.	deg/s

MAV_CMD_DO_VTOL_TRANSITION (3000

)

[Command] Request VTOL transition

Param (:Label)	Description	Values
1: State	The target VTOL state. For normal transitions, only MAV_VTOL_STATE_MC and MAV_VTOL_STATE_FW can be used.	MAV_VTOL_STATE
2: Immediate	Force immediate transition to the specified MAV_VTOL_STATE. 1: Force immediate, 0: normal transition. Can be used, for example, to trigger an emergency "Quadchute". Caution: Can be dangerous/damage vehicle, depending on autopilot implementation of this command.	

MAV_CMD_ARM_AUTHORIZATION_REQUEST (3001

)

[Command] Request authorization to arm the vehicle to a external entity, the arm authorizer is responsible to request all data that is needs from the vehicle before authorize or deny the request. If approved the progress of command_ack message should be set with

period of time that this authorization is valid in seconds or in case it was denied it should be set with one of the reasons in ARM_AUTH_DENIED_REASON.

Param (:Label)	Description	Values
1: System ID	Vehicle system id, this way ground station can request arm authorization on behalf of any vehicle	*min:*0 *max:*255 *increment:*1

MAV_CMD_SET_GUIDED_SUBMODE_STANDARD (4000

)

[Command] This command sets the submode to standard guided when vehicle is in guided mode. The vehicle holds position and altitude and the user can input the desired velocities along all three axes.

Param (:Label)	Description

MAV_CMD_SET_GUIDED_SUBMODE_CIRCLE (4001

)

[Command] This command sets submode circle when vehicle is in guided mode. Vehicle flies along a circle facing the center of the circle. The user can input the velocity along the circle and change the radius. If no input is given the vehicle will hold position.

Param (:Label)	Description	Units
1: Radius	Radius of desired circle in CIRCLE_MODE	m
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Target latitude of center of circle in CIRCLE_MODE	degE7
6: Longitude	Target longitude of center of circle in CIRCLE_MODE	degE7

MAV_CMD_CONDITION_GATE (4501

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Command] Delay mission state machine until gate has been reached.

Param (:Label)	Description	Values	Units
1: Geometry	Geometry: 0: orthogonal to path between previous and next waypoint.	*min:*0 *increment:*1	
2: UseAltitude	Altitude: 0: ignore altitude	*min:*0 *max:*1 *increment:*1	
3	Empty		
4	Empty		
5: Latitude	Latitude		
6: Longitude	Longitude		
7: Altitude	Altitude		m

MAV_CMD_NAV_FENCE_RETURN_POINT (5000

)

[Command] Fence return point (there can only be one such point in a geofence definition). If rally points are supported they should be used instead.

Param (:Label)	Description	Units
1	Reserved	
2	Reserved	
3	Reserved	
4	Reserved	
5: Latitude	Latitude	
6: Longitude	Longitude	
7: Altitude	Altitude	m

MAV_CMD_NAV_FENCE_POLYGON_VERTEX_INCLUSION (5001

)

[Command] Fence vertex for an inclusion polygon (the polygon must not be self-intersecting). The vehicle must stay within this area. Minimum of 3 vertices required.

Param (:Label)	Description	Values
1: Vertex Count	Polygon vertex count	*min:*3 *increment:*1
2: Inclusion Group	Vehicle must be inside ALL inclusion zones in a single group, vehicle must be inside at least one group, must be the same for all points in each polygon	*min:*0 *increment:*1
3	Reserved	
4	Reserved	
5: Latitude	Latitude	
6: Longitude	Longitude	
7	Reserved	

MAV_CMD_NAV_FENCE_POLYGON_VERTEX_EXCLUSION (5002

)

[Command] Fence vertex for an exclusion polygon (the polygon must not be self-intersecting). The vehicle must stay outside this area. Minimum of 3 vertices required.

Param (:Label)	Description	Values
1: Vertex Count	Polygon vertex count	*min:*3 *increment:*1
2	Reserved	
3	Reserved	
4	Reserved	
5: Latitude	Latitude	
6: Longitude	Longitude	
7	Reserved	

MAV_CMD_NAV_FENCE_CIRCLE_INCLUSION (5003

)

[Command] Circular fence area. The vehicle must stay inside this area.

Param (:Label)	Description	Values	Units
1: Radius	Radius.		m
2: Inclusion Group	Vehicle must be inside ALL inclusion zones in a single group, vehicle must be inside at least one group	*min:*0 *increment:*1	
3	Reserved		
4	Reserved		
5: Latitude	Latitude		
6: Longitude	Longitude		
7	Reserved		

MAV_CMD_NAV_FENCE_CIRCLE_EXCLUSION (5004

)

[Command] Circular fence area. The vehicle must stay outside this area.

Param (:Label)	Description	Units
1: Radius	Radius.	m
2	Reserved	
3	Reserved	
4	Reserved	
5: Latitude	Latitude	
6: Longitude	Longitude	
7	Reserved	

MAV_CMD_NAV_RALLY_POINT (5100

)

[Command] Rally point. You can have multiple rally points defined.

Param (:Label)	Description	Units
1	Reserved	
2	Reserved	
3	Reserved	
4	Reserved	
5: Latitude	Latitude	
6: Longitude	Longitude	
7: Altitude	Altitude	m

MAV_CMD_UAVCAN_GET_NODE_INFO (5200

)

[Command] Commands the vehicle to respond with a sequence of messages UAVCAN_NODE_INFO, one message per every UAVCAN node that is online. Note that some of the response messages can be lost, which the receiver can detect easily by checking whether every received UAVCAN_NODE_STATUS has a matching message UAVCAN_NODE_INFO received earlier; if not, this command should be sent again in order to request re-transmission of the node information messages.

Param (:Label)	Description
1	Reserved (set to 0)
2	Reserved (set to 0)
3	Reserved (set to 0)
4	Reserved (set to 0)
5	Reserved (set to 0)
6	Reserved (set to 0)
7	Reserved (set to 0)

MAV_CMD_DO_ADSB_OUT_IDENT (10001

)

[Command] Trigger the start of an ADSB-out IDENT. This should only be used when requested to do so by an Air Traffic Controller in controlled airspace. This starts the IDENT which is then typically held for 18 seconds by the hardware per the Mode A, C, and S transponder spec.

Param (:Label)	Description
1	Reserved (set to 0)
2	Reserved (set to 0)

3	Reserved (set to 0)
4	Reserved (set to 0)
5	Reserved (set to 0)
6	Reserved (set to 0)
7	Reserved (set to 0)

MAV_CMD_PAYLOAD_PREPARE_DEPLOY (30001

)

DEPRECATED: Replaced by (2021-06).

[Command] Deploy payload on a Lat / Lon / Alt position. This includes the navigation to reach the required release position and velocity.

Param (:Label)	Description	Values	Units
1: Operation Mode	Operation mode. 0: prepare single payload deploy (overwriting previous requests), but do not execute it. 1: execute payload deploy immediately (rejecting further deploy commands during execution, but allowing abort). 2: add payload deploy to existing deployment list.	*min:*0 *max:*2 *increment:*1	
2: Approach Vector	Desired approach vector in compass heading. A negative value indicates the system can define the approach vector at will.	*min:* -1 *max:*360	deg
3: Ground Speed	Desired ground speed at release time. This can be overridden by the airframe in case it needs to meet minimum airspeed. A negative value indicates the system can define the ground speed at will.	*min:* -1	
4: Altitude Clearance	Minimum altitude clearance to the release position. A negative value indicates the system can define the clearance at will.	*min:* -1	m
5: Latitude	Latitude. Note, if used in MISSION_ITEM (deprecated) the units are degrees (unscaled)		degE7
6: Longitude	Longitude. Note, if used in MISSION_ITEM (deprecated) the units are degrees (unscaled)		degE7
7: Altitude	Altitude (MSL)		m

MAV_CMD_PAYLOAD_CONTROL_DEPLOY (30002

)

DEPRECATED: Replaced by (2021-06).

[Command] Control the payload deployment.

Param (:Label)	Description	Values
1: Operation Mode	Operation mode. 0: Abort deployment, continue normal mission. 1: switch to payload deployment mode. 100: delete first payload deployment request. 101: delete all payload deployment requests.	*min:*0 *max:*101 *increment:*1
2	Reserved	
3	Reserved	
4	Reserved	
5	Reserved	
6	Reserved	
7	Reserved	

MAV_CMD_FIXED_MAG_CAL_YAW (42006

)

[Command] Magnetometer calibration based on provided known yaw. This allows for fast calibration using WMM field tables in the vehicle, given only the known yaw of the vehicle. If Latitude and longitude are both zero then use the current vehicle location.

Param (:Label)	Description	Units
1: Yaw	Yaw of vehicle in earth frame.	deg
2: CompassMask	CompassMask, 0 for all.	
3: Latitude	Latitude.	deg
4: Longitude	Longitude.	deg
5	Empty.	
6	Empty.	
7	Empty.	

MAV_CMD_DO_WINCH (42600

)

[Command] Command to operate winch.

Param (:Label)	Description	Values	Units
1: Instance	Winch instance number.	*min:*1 *increment:*1	
2: Action	Action to perform.	WINCH_ACTIONS	
3: Length	Length of line to release (negative to wind).		m
4: Rate	Release rate (negative to wind).		m/s
5	Empty.		
6	Empty.		

7	Empty.

MAV_CMD_WAYPOINT_USER_1 (31000

)

[Command] User defined waypoint item. Ground Station will show the Vehicle as flying through this item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_WAYPOINT_USER_2 (31001

)

[Command] User defined waypoint item. Ground Station will show the Vehicle as flying through this item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_WAYPOINT_USER_3 (31002

)

[Command] User defined waypoint item. Ground Station will show the Vehicle as flying through this item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_WAYPOINT_USER_4 (31003

)

[Command] User defined waypoint item. Ground Station will show the Vehicle as flying through this item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_WAYPOINT_USER_5 (31004

)

[Command] User defined waypoint item. Ground Station will show the Vehicle as flying through this item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_SPATIAL_USER_1 (31005

)

[Command] User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	

3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_SPATIAL_USER_2 (31006

)

[Command] User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_SPATIAL_USER_3 (31007

)

[Command] User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_SPATIAL_USER_4 (31008

)

[Command] User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_SPATIAL_USER_5 (31009

)

[Command] User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.

Param (:Label)	Description	Units
1	User defined	
2	User defined	
3	User defined	
4	User defined	
5: Latitude	Latitude unscaled	
6: Longitude	Longitude unscaled	
7: Altitude	Altitude (MSL)	m

MAV_CMD_USER_1 (31010

)

[Command] User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.

Param (:Label)	Description
1	User defined
2	User defined
3	User defined
4	User defined
5	User defined
6	User defined
7	User defined

MAV_CMD_USER_2 (31011

)

[Command] User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.

Param (:Label)	Description
1	User defined
2	User defined
3	User defined
4	User defined
5	User defined
6	User defined
7	User defined

MAV_CMD_USER_3 (31012

)

[Command] User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.

Param (:Label)	Description
1	User defined
2	User defined
3	User defined
4	User defined
5	User defined
6	User defined
7	User defined

MAV_CMD_USER_4 (31013

)

[Command] User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.

Param (:Label)	Description
1	User defined
2	User defined
3	User defined
4	User defined
5	User defined
6	User defined
7	User defined

MAV_CMD_USER_5 (31014

)

[Command] User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.

Param (:Label)	Description
1	User defined
2	User defined
3	User defined
4	User defined
5	User defined
6	User defined
7	User defined

MAV_CMD_CAN_FORWARD (32000

)

[Command] Request forwarding of CAN packets from the given CAN bus to this component. CAN Frames are sent using CAN_FRAME and CANFD_FRAME messages

Param (:Label)	Description
1: bus	Bus number (0 to disable forwarding, 1 for first bus, 2 for 2nd bus, 3 for 3rd bus).
2	Empty.
3	Empty.
4	Empty.
5	Empty.
6	Empty.
7	Empty.

MAVLink Messages

SYS_STATUS (#1

)

[Message] The general system state. If the system is following the MAVLink standard, the system state is mainly defined by three orthogonal states/modes: The system mode, which is either LOCKED (motors shut down and locked), MANUAL (system under RC control), GUIDED (system with autonomous position control, position setpoint controlled manually) or AUTO (system guided by path/waypoint planner). The NAV_MODE defined the current flight state: LIFTOFF (often an open-loop maneuver), LANDING,

WAYPOINTS or VECTOR. This represents the internal navigation state machine. The system status shows whether the system is currently active or not and if an emergency occurred. During the CRITICAL and EMERGENCY states the MAV is still considered to be active, but should start emergency procedures autonomously. After a failure occurred it should first move from active to critical to allow manual intervention and then move to emergency after a certain timeout.

Field Name	Type	Units	Values	Description
onboard_control_sensors_present	uint32_t		MAV_SYS_STATUS_SENSOR	Bitmap showing which onboard controllers and sensors are present. Value of 0: not present. Value of 1: present.
onboard_control_sensors_enabled	uint32_t		MAV_SYS_STATUS_SENSOR	Bitmap showing which onboard controllers and sensors are enabled: Value of 0: not enabled. Value of 1: enabled.
onboard_control_sensors_health	uint32_t		MAV_SYS_STATUS_SENSOR	Bitmap showing which onboard controllers and sensors have an error (or are operational). Value of 0: error. Value of 1: healthy.
load	uint16_t	d%		Maximum usage in percent of the mainloop time. Values: [0-1000] - should always be below 1000
voltage_battery	uint16_t	mV		Battery voltage, UINT16_MAX: Voltage not sent by autopilot
current_battery	int16_t	cA		Battery current, -1: Current not sent by autopilot
battery_remaining %	int8_t Battery energy remaining, -1: Battery remaining energy not sent by autopilot			
drop_rate_comm	uint16_t	c%		Communication drop rate, (UART, I2C, SPI, CAN), dropped packets on all links (packets that were corrupted on reception on the MAV)
errors_comm	uint16_t			Communication errors (UART, I2C, SPI, CAN), dropped packets on all links (packets that were corrupted on reception on the MAV)
errors_count1	uint16_t			Autopilot-specific errors
errors_count2	uint16_t			Autopilot-specific errors
errors_count3	uint16_t			Autopilot-specific errors
errors_count4	uint16_t			Autopilot-specific errors
				Bitmap showing which onboard

onboard_control_sensors_present_extended**	uint32_t		MAV_SYS_STATUS_SENSOR_EXTENDED	Bitmap showing which onboard controllers and sensors are present. Value of 0: not present. Value of 1: present.
onboard_control_sensors_enabled_extended**	uint32_t		MAV_SYS_STATUS_SENSOR_EXTENDED	Bitmap showing which onboard controllers and sensors are enabled: Value of 0: not enabled. Value of 1: enabled.
onboard_control_sensors_health_extended**	uint32_t		MAV_SYS_STATUS_SENSOR_EXTENDED	Bitmap showing which onboard controllers and sensors have an error (or are operational). Value of 0: error. Value of 1: healthy.

SYSTEM_TIME (#2

)

[Message] The system time is the time of the master clock, typically the computer clock of the main onboard computer.

Field Name	Type	Units	Description
time_unix_usec	uint64_t	us	Timestamp (UNIX epoch time).
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).

PING (#4

)

DEPRECATED: Replaced by SYSTEM_TIME (2011-08).

to be removed / merged with SYSTEM_TIME

[Message] A ping message either requesting or responding to a ping. This allows to measure the system latencies, including serial port, radio modem and UDP connections. The ping microservice is documented at https://mavlink.io/en/services/ping.html

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
seq	uint32_t		PING sequence
target_system	uint8_t		0: request ping from all receiving systems. If greater than 0: message is a ping response and number is the system id of the requesting system
target_component	uint8_t		0: request ping from all receiving components. If greater than 0: message is a ping response and number is the component id of the requesting component.

CHANGE_OPERATOR_CONTROL (#5

)

[Message] Request to control this MAV

Field Name	Type	Units	Description
target_system	uint8_t		System the GCS requests control for
control_request	uint8_t		0: request control of this MAV, 1: Release control of this MAV
version	uint8_t	rad	0: key as plaintext, 1-255: future, different hashing/encryption variants. The GCS should in general use the safest mode possible initially and then gradually move down the encryption level if it gets a NACK message indicating an encryption mismatch.
passkey	char[25]		Password / Key, depending on version plaintext or encrypted. 25 or less characters, NULL terminated. The characters may involve A-Z, a-z, 0-9, and "!?,.-"

CHANGE_OPERATOR_CONTROL_ACK (#6

)

[Message] Accept / deny control of this MAV

Field Name	Type	Description
gcs_system_id	uint8_t	ID of the GCS this message
control_request	uint8_t	0: request control of this MAV, 1: Release control of this MAV
ack	uint8_t	0: ACK, 1: NACK: Wrong passkey, 2: NACK: Unsupported passkey encryption method, 3: NACK: Already under control

AUTH_KEY (#7

)

[Message] Emit an encrypted signature / key identifying this system. PLEASE NOTE: This protocol has been kept simple, so transmitting the key requires an encrypted channel for true safety.

Field Name	Type	Description

key	char[32]	key

LINK_NODE_STATUS (#8

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] Status generated in each node in the communication chain and injected into MAVLink stream.

Field Name	Type	Units	Description
timestamp	uint64_t	ms	Timestamp (time since system boot).
tx_buf	uint8_t		
%			
Remaining free transmit buffer space			
rx_buf	uint8_t		
%			
Remaining free receive buffer space			
tx_rate	uint32_t	bytes/s	Transmit rate
rx_rate	uint32_t	bytes/s	Receive rate
rx_parse_err	uint16_t	bytes	Number of bytes that could not be parsed correctly.
tx_overflows	uint16_t	bytes	Transmit buffer overflows. This number wraps around as it reaches UINT16_MAX
rx_overflows	uint16_t	bytes	Receive buffer overflows. This number wraps around as it reaches UINT16_MAX
messages_sent	uint32_t		Messages sent
messages_received	uint32_t		Messages received (estimated from counting seq)
messages_lost	uint32_t		Messages lost (estimated from counting seq)

SET_MODE (#11

)

DEPRECATED: Replaced by MAV_CMD_DO_SET_MODE (2015-12).

Use COMMAND_LONG with MAV_CMD_DO_SET_MODE instead

[Message] Set the system mode, as defined by enum MAV_MODE. There is no target component id as the mode is by definition for the overall aircraft, not only for one component.

Field Name	Type	Values	Description
target_system	uint8_t		The system setting the mode
base_mode	uint8_t	MAV_MODE	The new base mode.
custom_mode	uint32_t		The new autopilot-specific mode. This field can be ignored by an autopilot.

PARAM_REQUEST_READ (#20

)

[Message] Request to read the onboard parameter with the param_id string id. Onboard parameters are stored as key[const char*] -> value[float]. This allows to send a parameter to any other component (such as the GCS) without the need of previous knowledge of possible parameter names. Thus the same GCS can store different parameters for different autopilots. See also https://mavlink.io/en/services/parameter.html for a full documentation of QGroundControl and IMU code.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
param_id	char[16]	Onboard parameter id, terminated by NULL if the length is less than 16 human-readable chars and WITHOUT null termination (NULL) byte if the length is exactly 16 chars - applications have to provide 16+1 bytes storage if the ID is stored as string
param_index	int16_t	Parameter index. Send -1 to use the param ID field as identifier (else the param id will be ignored)

PARAM_REQUEST_LIST (#21

)

[Message] Request all parameters of this component. After this request, all parameters are emitted. The parameter microservice is documented at https://mavlink.io/en/services/parameter.html

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID

PARAM_VALUE (#22

)

[Message] Emit the value of a onboard parameter. The inclusion of param_count and param_index in the message allows the recipient to keep track of received parameters and allows him to re-request missing parameters after a loss or timeout. The parameter microservice is documented at https://mavlink.io/en/services/parameter.html

Field Name	Type	Values	Description
param_id	char[16]		Onboard parameter id, terminated by NULL if the length is less than 16 human-readable chars and WITHOUT null termination (NULL) byte if the length is exactly 16 chars - applications have to provide 16+1 bytes storage if the ID is stored as string
param_value	float		Onboard parameter value
param_type	uint8_t	MAV_PARAM_TYPE	Onboard parameter type.
param_count	uint16_t		Total number of onboard parameters
param_index	uint16_t		Index of this onboard parameter

PARAM_SET (#23

)

[Message] Set a parameter value (write new value to permanent storage). The receiving component should acknowledge the new parameter value by broadcasting a PARAM_VALUE message (broadcasting ensures that multiple GCS all have an up-to-date list of all parameters). If the sending GCS did not receive a PARAM_VALUE within its timeout time, it should re-send the PARAM_SET message. The parameter microservice is documented at https://mavlink.io/en/services/parameter.html. PARAM_SET may also be called within the context of a transaction (started with MAV_CMD_PARAM_TRANSACTION). Within a transaction the receiving component should respond with PARAM_ACK_TRANSACTION to the setter component (instead of broadcasting PARAM_VALUE), and PARAM_SET should be re-sent if this is ACK not received.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
param_id	char[16]		Onboard parameter id, terminated by NULL if the length is less than 16 human-readable chars and WITHOUT null termination (NULL) byte if the length is exactly 16 chars - applications have to provide 16+1 bytes storage if the ID is stored as string
param_value	float		Onboard parameter value
param_type	uint8_t	MAV_PARAM_TYPE	Onboard parameter type.

GPS_RAW_INT (#24

)

[Message] The global position, as returned by the Global Positioning System (GPS). This is NOT the global position estimate of the system, but rather a RAW sensor value. See message GLOBAL_POSITION_INT for the global position estimate.

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
fix_type	uint8_t		GPS_FIX_TYPE	GPS fix type.
lat	int32_t	degE7		Latitude (WGS84, EGM96 ellipsoid)
lon	int32_t	degE7		Longitude (WGS84, EGM96 ellipsoid)
alt	int32_t	mm		Altitude (MSL). Positive for up. Note that virtually all GPS modules provide the MSL altitude in addition to the WGS84 altitude.
eph	uint16_t			GPS HDOP horizontal dilution of position (unitless * 100). If unknown, set to: UINT16_MAX
epv	uint16_t			GPS VDOP vertical dilution of position (unitless * 100). If unknown, set to: UINT16_MAX
vel	uint16_t	cm/s		GPS ground speed. If unknown, set to: UINT16_MAX
cog	uint16_t	cdeg		Course over ground (NOT heading, but direction of movement) in degrees * 100, 0.0..359.99 degrees. If unknown, set to: UINT16_MAX
satellites_visible	uint8_t			Number of satellites visible. If unknown, set to UINT8_MAX
alt_ellipsoid**	int32_t	mm		Altitude (above WGS84, EGM96 ellipsoid). Positive for up.
h_acc**	uint32_t	mm		Position uncertainty.
v_acc**	uint32_t	mm		Altitude uncertainty.
vel_acc**	uint32_t	mm		Speed uncertainty.
hdg_acc**	uint32_t	degE5		Heading / track uncertainty
yaw**	uint16_t	cdeg		Yaw in earth frame from north. Use 0 if this GPS does not provide yaw. Use UINT16_MAX if this GPS is configured to provide yaw and is currently unable to provide it. Use 36000 for north.

GPS_STATUS (#25

)

[Message] The positioning status, as reported by GPS. This message is intended to display status information about each satellite visible to the receiver. See message GLOBAL_POSITION_INT for the global position estimate. This message can contain information for up to 20 satellites.

Field Name	Type	Units	Description
satellites_visible	uint8_t		Number of satellites visible
satellite_prn	uint8_t[20]		Global satellite ID
satellite_used	uint8_t[20]		0: Satellite not used, 1: used for localization
satellite_elevation	uint8_t[20]	deg	Elevation (0: right on top of receiver, 90: on the horizon) of satellite
satellite_azimuth	uint8_t[20]	deg	Direction of satellite, 0: 0 deg, 255: 360 deg.
satellite_snr	uint8_t[20]	dB	Signal to noise ratio of satellite

SCALED_IMU (#26

)

[Message] The RAW IMU readings for the usual 9DOF sensor setup. This message should contain the scaled values to the described units

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
xacc	int16_t	mG	X acceleration
yacc	int16_t	mG	Y acceleration
zacc	int16_t	mG	Z acceleration
xgyro	int16_t	mrad/s	Angular speed around X axis
ygyro	int16_t	mrad/s	Angular speed around Y axis
zgyro	int16_t	mrad/s	Angular speed around Z axis
xmag	int16_t	mgauss	X Magnetic field
ymag	int16_t	mgauss	Y Magnetic field
zmag	int16_t	mgauss	Z Magnetic field

temperature**	int16_t	cdegC	Temperature, 0: IMU does not provide temperature values. If the IMU is at 0C it must send 1 (0.01C).

RAW_IMU (#27

)

[Message] The RAW IMU readings for a 9DOF sensor, which is identified by the id (default IMU1). This message should always contain the true raw values without any scaling to allow data capture and system debugging.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
xacc	int16_t		X acceleration (raw)
yacc	int16_t		Y acceleration (raw)
zacc	int16_t		Z acceleration (raw)
xgyro	int16_t		Angular speed around X axis (raw)
ygyro	int16_t		Angular speed around Y axis (raw)
zgyro	int16_t		Angular speed around Z axis (raw)
xmag	int16_t		X Magnetic field (raw)
ymag	int16_t		Y Magnetic field (raw)
zmag	int16_t		Z Magnetic field (raw)
id**	uint8_t		Id. Ids are numbered from 0 and map to IMUs numbered from 1 (e.g. IMU1 will have a message with id=0)
temperature**	int16_t	cdegC	Temperature, 0: IMU does not provide temperature values. If the IMU is at 0C it must send 1 (0.01C).

RAW_PRESSURE (#28

)

[Message] The RAW pressure readings for the typical setup of one absolute pressure and one differential pressure sensor. The sensor values should be the raw, UNSCALED ADC values.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
press_abs	int16_t		Absolute pressure (raw)
press_diff1	int16_t		Differential pressure 1 (raw, 0 if nonexistent)
press_diff2	int16_t		Differential pressure 2 (raw, 0 if nonexistent)
temperature	int16_t		Raw Temperature measurement (raw)

SCALED_PRESSURE (#29

)

[Message] The pressure readings for the typical setup of one absolute and differential pressure sensor. The units are as specified in each field.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
press_abs	float	hPa	Absolute pressure
press_diff	float	hPa	Differential pressure 1
temperature	int16_t	cdegC	Absolute pressure temperature
temperature_press_diff**	int16_t	cdegC	Differential pressure temperature (0, if not available). Report values of 0 (or 1) as 1 cdegC.

ATTITUDE (#30

)

[Message] The attitude in the aeronautical frame (right-handed, Z-down, X-front, Y-right).

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
roll	float	rad	Roll angle (-pi..+pi)
pitch	float	rad	Pitch angle (-pi..+pi)
yaw	float	rad	Yaw angle (-pi..+pi)
rollspeed	float	rad/s	Roll angular speed
pitchspeed	float	rad/s	Pitch angular speed
yawspeed	float	rad/s	Yaw angular speed

ATTITUDE_QUATERNION (#31

)

[Message] The attitude in the aeronautical frame (right-handed, Z-down, X-front, Y-right), expressed as quaternion. Quaternion order is w, x, y, z and a zero rotation would be expressed as (1 0 0 0).

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
q1	float		Quaternion component 1, w (1 in null-rotation)
q2	float		Quaternion component 2, x (0 in null-rotation)
q3	float		Quaternion component 3, y (0 in null-rotation)
q4	float		Quaternion component 4, z (0 in null-rotation)
rollspeed	float	rad/s	Roll angular speed
pitchspeed	float	rad/s	Pitch angular speed
yawspeed	float	rad/s	Yaw angular speed

repr_offset_q**	float[4]	Rotation offset by which the attitude quaternion and angular speed vector should be rotated for user display (quaternion with [w, x, y, z] order, zero-rotation is [1, 0, 0, 0], send [0, 0, 0, 0] if field not supported). This field is intended for systems in which the reference attitude may change during flight. For example, tailsitters VTOLs rotate their reference attitude by 90 degrees between hover mode and fixed wing mode, thus repr_offset_q is equal to [1, 0, 0, 0] in hover mode and equal to [0.7071, 0, 0.7071, 0] in fixed wing mode.

LOCAL_POSITION_NED (#32

)

[Message] The filtered local position (e.g. fused computer vision and accelerometers). Coordinate frame is right-handed, Z-axis down (aeronautical frame, NED / north-east-down convention)

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
x	float	m	X Position
y	float	m	Y Position
z	float	m	Z Position
vx	float	m/s	X Speed
vy	float	m/s	Y Speed
vz	float	m/s	Z Speed

GLOBAL_POSITION_INT (#33

)

[Message] The filtered global position (e.g. fused GPS and accelerometers). The position is in GPS-frame (right-handed, Z-up). It is designed as scaled integer message since the resolution of float is not sufficient.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
lat	int32_t	degE7	Latitude, expressed
lon	int32_t	degE7	Longitude, expressed
alt	int32_t	mm	Altitude (MSL). Note that virtually all GPS modules provide both WGS84 and MSL.
relative_alt	int32_t	mm	Altitude above ground
vx	int16_t	cm/s	Ground X Speed (Latitude, positive north)
vy	int16_t	cm/s	Ground Y Speed (Longitude, positive east)
vz	int16_t	cm/s	Ground Z Speed (Altitude, positive down)
hdg	uint16_t	cdeg	Vehicle heading (yaw angle), 0.0..359.99 degrees. If unknown, set to: UINT16_MAX

RC_CHANNELS_SCALED (#34

)

[Message] The scaled values of the RC channels received: (-100%) -10000, (0%) 0, (100%) 10000. Channels that are inactive should be set to UINT16_MAX.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
port	uint8_t		Servo output port (set of 8 outputs = 1 port). Flight stacks running on Pixhawk should use: 0 = MAIN, 1 = AUX.
chan1_scaled	int16_t		RC channel 1 value scaled.
chan2_scaled	int16_t		RC channel 2 value scaled.
chan3_scaled	int16_t		RC channel 3 value scaled.
chan4_scaled	int16_t		RC channel 4 value scaled.
chan5_scaled	int16_t		RC channel 5 value scaled.
chan6_scaled	int16_t		RC channel 6 value scaled.
chan7_scaled	int16_t		RC channel 7 value scaled.
chan8_scaled	int16_t		RC channel 8 value scaled.
rssi	uint8_t		Receive signal strength indicator in device-dependent units/scale. Values: [0-254], UINT8_MAX: invalid/unknown.

RC_CHANNELS_RAW (#35

)

[Message] The RAW values of the RC channels received. The standard PPM modulation is as follows: 1000 microseconds: 0%, 2000 microseconds: 100%. A value of UINT16_MAX implies the channel is unused. Individual receivers/transmitters might violate this specification.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
port	uint8_t		Servo output port (set of 8 outputs = 1 port). Flight stacks running on Pixhawk should use: 0 = MAIN, 1 = AUX.
chan1_raw	uint16_t	us	RC channel 1 value.
chan2_raw	uint16_t	us	RC channel 2 value.
chan3_raw	uint16_t	us	RC channel 3 value.
chan4_raw	uint16_t	us	RC channel 4 value.
chan5_raw	uint16_t	us	RC channel 5 value.
chan6_raw	uint16_t	us	RC channel 6 value.
chan7_raw	uint16_t	us	RC channel 7 value.
chan8_raw	uint16_t	us	RC channel 8 value.
rssi	uint8_t		Receive signal strength indicator in device-dependent units/scale. Values: [0-254], UINT8_MAX: invalid/unknown.

SERVO_OUTPUT_RAW (#36

)

[Message] Superseded by ACTUATOR_OUTPUT_STATUS. The RAW values of the servo outputs (for RC input from the remote, use the RC_CHANNELS messages). The standard PPM modulation is as follows: 1000 microseconds: 0%, 2000 microseconds: 100%.

Field Name	Type	Units	Description
time_usec	uint32_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
port	uint8_t		Servo output port (set of 8 outputs = 1 port). Flight stacks running on Pixhawk should use: 0 = MAIN, 1 = AUX.
servo1_raw	uint16_t	us	Servo output 1 value
servo2_raw	uint16_t	us	Servo output 2 value
servo3_raw	uint16_t	us	Servo output 3 value
servo4_raw	uint16_t	us	Servo output 4 value
servo5_raw	uint16_t	us	Servo output 5 value
servo6_raw	uint16_t	us	Servo output 6 value
servo7_raw	uint16_t	us	Servo output 7 value
servo8_raw	uint16_t	us	Servo output 8 value
servo9_raw**	uint16_t	us	Servo output 9 value
servo10_raw**	uint16_t	us	Servo output 10 value
servo11_raw**	uint16_t	us	Servo output 11 value
servo12_raw**	uint16_t	us	Servo output 12 value
servo13_raw**	uint16_t	us	Servo output 13 value
servo14_raw**	uint16_t	us	Servo output 14 value
servo15_raw**	uint16_t	us	Servo output 15 value
servo16_raw**	uint16_t	us	Servo output 16 value

MISSION_REQUEST_PARTIAL_LIST (#37

)

[Message] Request a partial list of mission items from the system/component. https://mavlink.io/en/services/mission.html. If start and end index are the same, just send one waypoint.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
start_index	int16_t		Start index
end_index	int16_t		End index, -1 by default (-1: send list to end). Else a valid index of the list
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

MISSION_WRITE_PARTIAL_LIST (#38

)

[Message] This message is sent to the MAV to write a partial list. If start index == end index, only one item will be transmitted / updated. If the start index is NOT 0 and above the current list size, this request should be REJECTED!

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
start_index	int16_t		Start index. Must be smaller / equal to the largest index of the current onboard list.
end_index	int16_t		End index, equal or greater than start index.
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

MISSION_ITEM (#39

)

DEPRECATED: Replaced by MISSION_ITEM_INT (2020-06).

[Message] Message encoding a mission item. This message is emitted to announce the presence of a mission item and to set a mission item on the system. The mission item can be either in x, y, z meters (type: LOCAL) or x:lat, y:lon, z:altitude. Local frame is Z-down, right handed (NED), global frame is Z-up, right handed (ENU). NaN may be used to indicate an optional/default value (e.g. to use the system's current latitude or yaw rather than a specific value). See also https://mavlink.io/en/services/mission.html.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
seq	uint16_t		Sequence
frame	uint8_t	MAV_FRAME	The coordinate system of the waypoint.
command	uint16_t	MAV_CMD	The scheduled action for the waypoint.
current	uint8_t		false:0, true:1
autocontinue	uint8_t		Autocontinue to next waypoint
param1	float		PARAM1, see MAV_CMD enum
param2	float		PARAM2, see MAV_CMD enum
param3	float		PARAM3, see MAV_CMD enum
param4	float		PARAM4, see MAV_CMD enum
x	float		PARAM5 / local: X coordinate, global: latitude
y	float		PARAM6 / local: Y coordinate, global: longitude
z	float		PARAM7 / local: Z coordinate, global: altitude (relative or absolute, depending on frame).
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

MISSION_REQUEST (#40

)

DEPRECATED: Replaced by MISSION_REQUEST_INT (2020-06).

A system that gets this request should respond with MISSION_ITEM_INT (as though MISSION_REQUEST_INT was received).

[Message] Request the information of the mission item with the sequence number seq. The response of the system to this message should be a MISSION_ITEM message. https://mavlink.io/en/services/mission.html

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
seq	uint16_t		Sequence
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

MISSION_SET_CURRENT (#41

)

[Message] Set the mission item with sequence number seq as current item. This means that the MAV will continue to this mission item on the shortest path (not following the mission items in-between).

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
seq	uint16_t	Sequence

MISSION_CURRENT (#42

)

[Message] Message that announces the sequence number of the current active mission item. The MAV will fly towards this mission item.

Field Name	Type	Description
seq	uint16_t	Sequence

MISSION_REQUEST_LIST (#43

)

[Message] Request the overall list of mission items from the system/component.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

MISSION_COUNT (#44

)

[Message] This message is emitted as response to MISSION_REQUEST_LIST by the MAV and to initiate a write transaction. The GCS can then request the individual mission item based on the knowledge of the total number of waypoints.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
count	uint16_t		Number of mission items in the sequence
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

MISSION_CLEAR_ALL (#45

)

[Message] Delete all mission items at once.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

MISSION_ITEM_REACHED (#46

)

[Message] A certain mission item has been reached. The system will either hold this position (or circle on the orbit) or (if the autocontinue on the WP was set) continue to the next waypoint.

Field Name	Type	Description
seq	uint16_t	Sequence

MISSION_ACK (#47

)

[Message] Acknowledgment message during waypoint handling. The type field states if this message is a positive ack (type=0) or if an error happened (type=non-zero).

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
type	uint8_t	MAV_MISSION_RESULT	Mission result.
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

SET_GPS_GLOBAL_ORIGIN (#48

)

[Message] Sets the GPS coordinates of the vehicle local origin (0,0,0) position. Vehicle should emit GPS_GLOBAL_ORIGIN irrespective of whether the origin is changed. This enables transform between the local coordinate frame and the global (GPS) coordinate frame, which may be necessary when (for example) indoor and outdoor settings are connected and the MAV should move from in- to outdoor.

Field Name	Type	Units	Description
target_system	uint8_t		System ID
latitude	int32_t	degE7	Latitude (WGS84)
longitude	int32_t	degE7	Longitude (WGS84)
altitude	int32_t	mm	Altitude (MSL). Positive for up.
time_usec**	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.

GPS_GLOBAL_ORIGIN (#49

)

[Message] Publishes the GPS coordinates of the vehicle local origin (0,0,0) position. Emitted whenever a new GPS-Local position mapping is requested or set - e.g. following SET_GPS_GLOBAL_ORIGIN message.

Field Name	Type	Units	Description
latitude	int32_t	degE7	Latitude (WGS84)
longitude	int32_t	degE7	Longitude (WGS84)
altitude	int32_t	mm	Altitude (MSL). Positive for up.
time_usec**	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.

PARAM_MAP_RC (#50

)

[Message] Bind a RC channel to a parameter. The parameter should change according to the RC channel value.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
param_id	char[16]	Onboard parameter id, terminated by NULL if the length is less than 16 human-readable chars and WITHOUT null termination (NULL) byte if the length is exactly 16 chars - applications have to provide 16+1 bytes storage if the ID is stored as string
param_index	int16_t	Parameter index. Send -1 to use the param ID field as identifier (else the param id will be ignored), send -2 to disable any existing map for this rc_channel_index.
parameter_rc_channel_index	uint8_t	Index of parameter RC channel. Not equal to the RC channel id. Typically corresponds to a potentiometer-knob on the RC.
param_value0	float	Initial parameter value
scale	float	Scale, maps the RC range [-1, 1] to a parameter value
param_value_min	float	Minimum param value. The protocol does not define if this overwrites an onboard minimum value. (Depends on implementation)
param_value_max	float	Maximum param value. The protocol does not define if this overwrites an onboard maximum value. (Depends on implementation)

MISSION_REQUEST_INT (#51

)

[Message] Request the information of the mission item with the sequence number seq. The response of the system to this message should be a MISSION_ITEM_INT message. https://mavlink.io/en/services/mission.html

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
seq	uint16_t		Sequence
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

SAFETY_SET_ALLOWED_AREA (#54

)

[Message] Set a safety zone (volume), which is defined by two corners of a cube. This message can be used to tell the MAV which setpoints/waypoints to accept and which to reject. Safety areas are often enforced by national or competition regulations.

Field Name	Type	Units	Values	Description
target_system	uint8_t			System ID
target_component	uint8_t			Component ID
frame	uint8_t		MAV_FRAME	Coordinate frame. Can be either global, GPS, right-handed with Z axis up or local, right handed, Z axis down.
p1x	float	m		x position 1 / Latitude 1
p1y	float	m		y position 1 / Longitude 1
p1z	float	m		z position 1 / Altitude 1

p2x	float	m	x position 2 / Latitude 2
p2y	float	m	y position 2 / Longitude 2
p2z	float	m	z position 2 / Altitude 2

SAFETY_ALLOWED_AREA (#55

)

[Message] Read out the safety zone the MAV currently assumes.

Field Name	Type	Units	Values	Description
frame	uint8_t		MAV_FRAME	Coordinate frame. Can be either global, GPS, right-handed with Z axis up or local, right handed, Z axis down.
p1x	float	m		x position 1 / Latitude 1
p1y	float	m		y position 1 / Longitude 1
p1z	float	m		z position 1 / Altitude 1
p2x	float	m		x position 2 / Latitude 2
p2y	float	m		y position 2 / Longitude 2
p2z	float	m		z position 2 / Altitude 2

ATTITUDE_QUATERNION_COV (#61

)

[Message] The attitude in the aeronautical frame (right-handed, Z-down, X-front, Y-right), expressed as quaternion. Quaternion order is w, x, y, z and a zero rotation would be expressed as (1 0 0 0).

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
q	float[4]		Quaternion components, w, x, y, z (1 0 0 0 is the null-rotation)
rollspeed	float	rad/s	Roll angular speed
pitchspeed	float	rad/s	Pitch angular speed
yawspeed	float	rad/s	Yaw angular speed
covariance	float[9]		Row-major representation of a 3x3 attitude covariance matrix (states: roll, pitch, yaw; first three entries are the first ROW, next three entries are the second row, etc.). If unknown, assign NaN value to first element in the array.

NAV_CONTROLLER_OUTPUT (#62

)

[Message] The state of the navigation and position controller.

Field Name	Type	Units	Description
nav_roll	float	deg	Current desired roll
nav_pitch	float	deg	Current desired pitch
nav_bearing	int16_t	deg	Current desired heading
target_bearing	int16_t	deg	Bearing to current waypoint/target
wp_dist	uint16_t	m	Distance to active waypoint
alt_error	float	m	Current altitude error
aspd_error	float	m/s	Current airspeed error
xtrack_error	float	m	Current crosstrack error on x-y plane

GLOBAL_POSITION_INT_COV (#63

)

[Message] The filtered global position (e.g. fused GPS and accelerometers). The position is in GPS-frame (right-handed, Z-up). It is designed as scaled integer message since the resolution of float is not sufficient. NOTE: This message is intended for onboard networks / companion computers and higher-bandwidth links and optimized for accuracy and completeness. Please use the GLOBAL_POSITION_INT message for a minimal subset.

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
estimator_type	uint8_t		MAV_ESTIMATOR_TYPE	Class id of the estimator this estimate originated from.
lat	int32_t	degE7		Latitude
lon	int32_t	degE7		Longitude
alt	int32_t	mm		Altitude in meters above MSL
relative_alt	int32_t	mm		Altitude above ground
vx	float	m/s		Ground X Speed (Latitude)
vy	float	m/s		Ground Y Speed (Longitude)
vz	float	m/s		Ground Z Speed (Altitude)
covariance	float[36]			Row-major representation of a 6x6 position and velocity 6x6 cross-covariance matrix (states: lat, lon, alt, vx, vy, vz; first six entries are the first ROW, next six entries are the second row, etc.). If unknown, assign NaN value to first element in the array.

LOCAL_POSITION_NED_COV (#64

)

[Message] The filtered local position (e.g. fused computer vision and accelerometers). Coordinate frame is right-handed, Z-axis down (aeronautical frame, NED / north-east-down convention)

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
estimator_type	uint8_t		MAV_ESTIMATOR_TYPE	Class id of the estimator this estimate originated from.
x	float	m		X Position
y	float	m		Y Position
z	float	m		Z Position
vx	float	m/s		X Speed
vy	float	m/s		Y Speed
vz	float	m/s		Z Speed
ax	float	m/s/s		X Acceleration
ay	float	m/s/s		Y Acceleration
az	float	m/s/s		Z Acceleration
covariance	float[45]			Row-major representation of position, velocity and acceleration 9x9 cross-covariance matrix upper right triangle (states: x, y, z, vx, vy, vz, ax, ay, az; first nine entries are the first ROW, next eight entries are the second row, etc.). If unknown, assign NaN value to first element in the array.

RC_CHANNELS (#65

)

[Message] The PPM values of the RC channels received. The standard PPM modulation is as follows: 1000 microseconds: 0%, 2000 microseconds: 100%. A value of UINT16_MAX implies the channel is unused. Individual receivers/transmitters might violate this specification.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
chancount	uint8_t		Total number of RC channels being received. This can be larger than 18, indicating that more channels are available but not given in this message. This value should be 0 when no RC channels are available.
chan1_raw	uint16_t	us	RC channel 1 value.
chan2_raw	uint16_t	us	RC channel 2 value.
chan3_raw	uint16_t	us	RC channel 3 value.
chan4_raw	uint16_t	us	RC channel 4 value.
chan5_raw	uint16_t	us	RC channel 5 value.
chan6_raw	uint16_t	us	RC channel 6 value.
chan7_raw	uint16_t	us	RC channel 7 value.
chan8_raw	uint16_t	us	RC channel 8 value.
chan9_raw	uint16_t	us	RC channel 9 value.
chan10_raw	uint16_t	us	RC channel 10 value.
chan11_raw	uint16_t	us	RC channel 11 value.
chan12_raw	uint16_t	us	RC channel 12 value.
chan13_raw	uint16_t	us	RC channel 13 value.
chan14_raw	uint16_t	us	RC channel 14 value.
chan15_raw	uint16_t	us	RC channel 15 value.
chan16_raw	uint16_t	us	RC channel 16 value.
chan17_raw	uint16_t	us	RC channel 17 value.
chan18_raw	uint16_t	us	RC channel 18 value.
rssi	uint8_t		Receive signal strength indicator in device-dependent units/scale. Values: [0-254], UINT8_MAX: invalid/unknown.

REQUEST_DATA_STREAM (#66

)

DEPRECATED: Replaced by SET_MESSAGE_INTERVAL (2015-08).

[Message] Request a data stream.

Field Name	Type	Units	Description
target_system	uint8_t		The target requested to send the message stream.
target_component	uint8_t		The target requested to send the message stream.
req_stream_id	uint8_t		The ID of the requested data stream
req_message_rate	uint16_t	Hz	The requested message rate
start_stop	uint8_t		1 to start sending, 0 to stop sending.

DATA_STREAM (#67

)

DEPRECATED: Replaced by MESSAGE_INTERVAL (2015-08).

[Message] Data stream status information.

Field Name	Type	Units	Description
stream_id	uint8_t		The ID of the requested data stream
message_rate	uint16_t	Hz	The message rate
on_off	uint8_t		1 stream is enabled, 0 stream is stopped.

MANUAL_CONTROL (#69

)

[Message] This message provides an API for manually controlling the vehicle using standard joystick axes nomenclature, along with a joystick-like input device. Unused axes can be disabled and buttons states are transmitted as individual on/off bits of a bitmask

Field Name	Type	Description
target	uint8_t	The system to be controlled.
x	int16_t	X-axis, normalized to the range [-1000,1000]. A value of INT16_MAX indicates that this axis is invalid. Generally corresponds to forward(1000)-backward(-1000) movement on a joystick and the pitch of a vehicle.
y	int16_t	Y-axis, normalized to the range [-1000,1000]. A value of INT16_MAX indicates that this axis is invalid. Generally corresponds to left(-1000)-right(1000) movement on a joystick and the roll of a vehicle.
z	int16_t	Z-axis, normalized to the range [-1000,1000]. A value of INT16_MAX indicates that this axis is invalid. Generally corresponds to a separate slider movement with maximum being 1000 and minimum being -1000 on a joystick and the thrust of a vehicle. Positive values are positive thrust, negative values are negative thrust.
r	int16_t	R-axis, normalized to the range [-1000,1000]. A value of INT16_MAX indicates that this axis is invalid. Generally corresponds to a twisting of the joystick, with counter-clockwise being 1000 and clockwise being -1000, and the yaw of a vehicle.
buttons	uint16_t	A bitfield corresponding to the joystick buttons' 0-15 current state, 1 for pressed, 0 for released. The lowest bit corresponds to Button 1.
buttons2**	uint16_t	A bitfield corresponding to the joystick buttons' 16-31 current state, 1 for pressed, 0 for released. The lowest bit corresponds to Button 16.
enabled_extensions**	uint8_t	Set bits to 1 to indicate which of the following extension fields contain valid data: bit 0: pitch, bit 1: roll.
s**	int16_t	Pitch-only-axis, normalized to the range [-1000,1000]. Generally corresponds to pitch on vehicles with additional degrees of freedom. Valid if bit 0 of enabled_extensions field is set. Set to 0 if invalid.
t**	int16_t	Roll-only-axis, normalized to the range [-1000,1000]. Generally corresponds to roll on vehicles with additional degrees of freedom. Valid if bit 1 of enabled_extensions field is set. Set to 0 if invalid.

RC_CHANNELS_OVERRIDE (#70

)

[Message] The RAW values of the RC channels sent to the MAV to override info received from the RC radio. The standard PPM modulation is as follows: 1000 microseconds: 0%, 2000 microseconds: 100%. Individual receivers/transmitters might violate this specification. Note carefully the semantic differences between the first 8 channels and the subsequent channels

Field Name	Type	Units	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
chan1_raw	uint16_t	us	RC channel 1 value. A value of UINT16_MAX means to ignore this field. A value of 0 means to release this channel back to the RC radio.
chan2_raw	uint16_t	us	RC channel 2 value. A value of UINT16_MAX means to ignore this field. A value of 0 means to release this channel back to the RC radio.
chan3_raw	uint16_t	us	RC channel 3 value. A value of UINT16_MAX means to ignore this field. A value of 0 means to release this channel back to the RC radio.
chan4_raw	uint16_t	us	RC channel 4 value. A value of UINT16_MAX means to ignore this field. A value of 0 means to release this channel back to the RC radio.
chan5_raw	uint16_t	us	RC channel 5 value. A value of UINT16_MAX means to ignore this field. A value of 0 means to release this channel back to the RC radio.
chan6_raw	uint16_t	us	RC channel 6 value. A value of UINT16_MAX means to ignore this field. A value of 0 means to release this channel back to the RC radio.
chan7_raw	uint16_t	us	RC channel 7 value. A value of UINT16_MAX means to ignore this field. A value of 0 means to release this channel back to the RC radio.
chan8_raw	uint16_t	us	RC channel 8 value. A value of UINT16_MAX means to ignore this field. A value of 0 means to release this channel back to the RC radio.
chan9_raw**	uint16_t	us	RC channel 9 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.
chan10_raw**	uint16_t	us	RC channel 10 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.
chan11_raw**	uint16_t	us	RC channel 11 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.
chan12_raw**	uint16_t	us	RC channel 12 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.
chan13_raw**	uint16_t	us	RC channel 13 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.
chan14_raw**	uint16_t	us	RC channel 14 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.
chan15_raw**	uint16_t	us	RC channel 15 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.
chan16_raw**	uint16_t	us	RC channel 16 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.
chan17_raw**	uint16_t	us	RC channel 17 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.
chan18_raw**	uint16_t	us	RC channel 18 value. A value of 0 or UINT16_MAX means to ignore this field. A value of UINT16_MAX-1 means to release this channel back to the RC radio.

MISSION_ITEM_INT (#73

)

[Message] Message encoding a mission item. This message is emitted to announce the presence of a mission item and to set a mission item on the system. The mission item can be either in x, y, z meters (type: LOCAL) or x:lat, y:lon, z:altitude. Local frame is Z-down, right handed (NED), global frame is Z-up, right handed (ENU). NaN or INT32_MAX may be used in float/integer params (respectively) to indicate optional/default values (e.g. to use the component's current latitude, yaw rather than a specific value). See also https://mavlink.io/en/services/mission.html.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID

seq	uint16_t		Waypoint ID (sequence number). Starts at zero. Increases monotonically for each waypoint, no gaps in the sequence (0,1,2,3,4).
frame	uint8_t	MAV_FRAME	The coordinate system of the waypoint.
command	uint16_t	MAV_CMD	The scheduled action for the waypoint.
current	uint8_t		false:0, true:1
autocontinue	uint8_t		Autocontinue to next waypoint
param1	float		PARAM1, see MAV_CMD enum
param2	float		PARAM2, see MAV_CMD enum
param3	float		PARAM3, see MAV_CMD enum
param4	float		PARAM4, see MAV_CMD enum
x	int32_t		PARAM5 / local: x position in meters * 1e4, global: latitude in degrees * 10^7
y	int32_t		PARAM6 / y position: local: x position in meters * 1e4, global: longitude in degrees *10^7
z	float		PARAM7 / z position: global: altitude in meters (relative or absolute, depending on frame.
mission_type**	uint8_t	MAV_MISSION_TYPE	Mission type.

VFR_HUD (#74

)

[Message] Metrics typically displayed on a HUD for fixed wing aircraft.

Field Name	Type	Units	Description
airspeed	float	m/s	Vehicle speed in form appropriate for vehicle type. For standard aircraft this is typically calibrated airspeed (CAS) or indicated airspeed (IAS) - either of which can be used by a pilot to estimate stall speed.
groundspeed	float	m/s	Current ground speed.
heading	int16_t	deg	Current heading in compass units (0-360, 0=north).
throttle	uint16_t	%	Current throttle setting (0 to 100).
alt	float	m	Current altitude (MSL).
climb	float	m/s	Current climb rate.

COMMAND_INT (#75

)

[Message] Message encoding a command with parameters as scaled integers. Scaling depends on the actual command value. NaN or INT32_MAX may be used in float/integer params (respectively) to indicate optional/default values (e.g. to use the component's current latitude, yaw rather than a specific value). The command microservice is documented at https://mavlink.io/en/services/command.html

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
frame	uint8_t	MAV_FRAME	The coordinate system of the COMMAND.
command	uint16_t	MAV_CMD	The scheduled action for the mission item.
current	uint8_t		Not used.
autocontinue	uint8_t		Not used (set 0).
param1	float		PARAM1, see MAV_CMD enum
param2	float		PARAM2, see MAV_CMD enum
param3	float		PARAM3, see MAV_CMD enum
param4	float		PARAM4, see MAV_CMD enum
x	int32_t		PARAM5 / local: x position in meters * 1e4, global: latitude in degrees * 10^7
y	int32_t		PARAM6 / local: y position in meters * 1e4, global: longitude in degrees * 10^7
z	float		PARAM7 / z position: global: altitude in meters (relative or absolute, depending on frame).

COMMAND_LONG (#76

)

[Message] Send a command with up to seven parameters to the MAV. The command microservice is documented at https://mavlink.io/en/services/command.html

Field Name	Type	Values	Description
target_system	uint8_t		System which should execute the command
target_component	uint8_t		Component which should execute the command, 0 for all components
command	uint16_t	MAV_CMD	Command ID (of command to send).
confirmation	uint8_t		0: First transmission of this command. 1-255: Confirmation transmissions (e.g. for kill command)
param1	float		Parameter 1 (for the specific command).
param2	float		Parameter 2 (for the specific command).
param3	float		Parameter 3 (for the specific command).
param4	float		Parameter 4 (for the specific command).
param5	float		Parameter 5 (for the specific command).
param6	float		Parameter 6 (for the specific command).
param7	float		Parameter 7 (for the specific command).

COMMAND_ACK (#77

)

[Message] Report status of a command. Includes feedback whether the command was executed. The command microservice is documented at https://mavlink.io/en/services/command.html

Field Name	Type	Values	Description
command	uint16_t	MAV_CMD	Command ID (of acknowledged command).
result	uint8_t	MAV_RESULT	Result of command.
progress**	uint8_t		Also used as result_param1, it can be set with an enum containing the errors reasons of why the command was denied, or the progress percentage when result is MAV_RESULT_IN_PROGRESS (UINT8_MAX if the progress is unknown).
result_param2**	int32_t		Additional parameter of the result, example: which parameter of MAV_CMD_NAV_WAYPOINT caused it to be denied.
target_system**	uint8_t		System ID of the target recipient. This is the ID of the system that sent the command for which this COMMAND_ACK is an acknowledgement.
target_component**	uint8_t		Component ID of the target recipient. This is the ID of the system that sent the command for which this COMMAND_ACK is an acknowledgement.

COMMAND_CANCEL (#80

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] Cancel a long running command. The target system should respond with a COMMAND_ACK to the original command with result=MAV_RESULT_CANCELLED if the long running process was cancelled. If it has already completed, the cancel action can be ignored. The cancel action can be retried until some sort of acknowledgement to the original command has been received. The command microservice is documented at https://mavlink.io/en/services/command.html

Field Name	Type	Values	Description
target_system	uint8_t		System executing long running command. Should not be broadcast (0).
target_component	uint8_t		Component executing long running command.
command	uint16_t	MAV_CMD	Command ID (of command to cancel).

MANUAL_SETPOINT (#81

)

[Message] Setpoint in roll, pitch, yaw and thrust from the operator

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
roll	float	rad/s	Desired roll rate
pitch	float	rad/s	Desired pitch rate
yaw	float	rad/s	Desired yaw rate
thrust	float		Collective thrust, normalized to 0 .. 1
mode_switch	uint8_t		Flight mode switch position, 0.. 255
manual_override_switch	uint8_t		Override mode switch position, 0.. 255

SET_ATTITUDE_TARGET (#82

)

[Message] Sets a desired vehicle attitude. Used by an external controller to command the vehicle (manual controller or other system).

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
target_system	uint8_t			System ID
target_component	uint8_t			Component ID
type_mask	uint8_t		ATTITUDE_TARGET_TYPEMASK	Bitmap to indicate which dimensions should be ignored by the vehicle.
q	float[4]			Attitude quaternion (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
body_roll_rate	float	rad/s		Body roll rate
body_pitch_rate	float	rad/s		Body pitch rate
body_yaw_rate	float	rad/s		Body yaw rate
thrust	float			Collective thrust, normalized to 0 .. 1 (-1 .. 1 for vehicles capable of reverse trust)
thrust_body**	float[3]			3D thrust setpoint in the body NED frame, normalized to -1 .. 1

ATTITUDE_TARGET (#83

)

[Message] Reports the current commanded attitude of the vehicle as specified by the autopilot. This should match the commands sent in a SET_ATTITUDE_TARGET message if the vehicle is being controlled this way.

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
type_mask	uint8_t		ATTITUDE_TARGET_TYPEMASK	Bitmap to indicate which dimensions should be ignored by the vehicle.
q	float[4]			Attitude quaternion (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
body_roll_rate	float	rad/s		Body roll rate
body_pitch_rate	float	rad/s		Body pitch rate
body_yaw_rate	float	rad/s		Body yaw rate
thrust	float			Collective thrust, normalized to 0 .. 1 (-1 .. 1 for vehicles capable of reverse trust)

SET_POSITION_TARGET_LOCAL_NED (#84

)

[Message] Sets a desired vehicle position in a local north-east-down coordinate frame. Used by an external controller to command the vehicle (manual controller or other system).

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
target_system	uint8_t			System ID
target_component	uint8_t			Component ID
coordinate_frame	uint8_t		MAV_FRAME	Valid options are: MAV_FRAME_LOCAL_NED = 1, MAV_FRAME_LOCAL_OFFSET_NED = 7, MAV_FRAME_BODY_NED = 8, MAV_FRAME_BODY_OFFSET_NED = 9
type_mask	uint16_t		POSITION_TARGET_TYPEMASK	Bitmap to indicate which dimensions should be ignored by the vehicle.
x	float	m		X Position in NED frame
y	float	m		Y Position in NED frame
z	float	m		Z Position in NED frame (note, altitude is negative in NED)
vx	float	m/s		X velocity in NED frame
vy	float	m/s		Y velocity in NED frame
vz	float	m/s		Z velocity in NED frame
afx	float	m/s/s		X acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afy	float	m/s/s		Y acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afz	float	m/s/s		Z acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
yaw	float	rad		yaw setpoint
yaw_rate	float	rad/s		yaw rate setpoint

POSITION_TARGET_LOCAL_NED (#85

)

[Message] Reports the current commanded vehicle position, velocity, and acceleration as specified by the autopilot. This should match the commands sent in SET_POSITION_TARGET_LOCAL_NED if the vehicle is being controlled this way.

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
coordinate_frame	uint8_t		MAV_FRAME	Valid options are: MAV_FRAME_LOCAL_NED = 1, MAV_FRAME_LOCAL_OFFSET_NED = 7, MAV_FRAME_BODY_NED = 8, MAV_FRAME_BODY_OFFSET_NED = 9
type_mask	uint16_t		POSITION_TARGET_TYPEMASK	Bitmap to indicate which dimensions should be ignored by the vehicle.
x	float	m		X Position in NED frame
y	float	m		Y Position in NED frame
z	float	m		Z Position in NED frame (note, altitude is negative in NED)
vx	float	m/s		X velocity in NED frame
vy	float	m/s		Y velocity in NED frame
vz	float	m/s		Z velocity in NED frame
afx	float	m/s/s		X acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afy	float	m/s/s		Y acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afz	float	m/s/s		Z acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
yaw	float	rad		yaw setpoint
yaw_rate	float	rad/s		yaw rate setpoint

SET_POSITION_TARGET_GLOBAL_INT (#86

)

[Message] Sets a desired vehicle position, velocity, and/or acceleration in a global coordinate system (WGS84). Used by an external controller to command the vehicle (manual controller or other system).

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot). The rationale for the timestamp in the setpoint is to allow the system to compensate for the transport delay of the setpoint. This allows the system to compensate processing latency.
target_system	uint8_t			System ID
target_component	uint8_t			Component ID
coordinate_frame	uint8_t		MAV_FRAME	Valid options are: MAV_FRAME_GLOBAL_INT = 5, MAV_FRAME_GLOBAL_RELATIVE_ALT_INT = 6, MAV_FRAME_GLOBAL_TERRAIN_ALT_INT = 11
type_mask	uint16_t		POSITION_TARGET_TYPEMASK	Bitmap to indicate which dimensions should be ignored by the vehicle.
lat_int	int32_t	degE7		X Position in WGS84 frame
lon_int	int32_t	degE7		Y Position in WGS84 frame
alt	float	m		Altitude (MSL, Relative to home, or AGL - depending on frame)
vx	float	m/s		X velocity in NED frame
vy	float	m/s		Y velocity in NED frame
vz	float	m/s		Z velocity in NED frame
afx	float	m/s/s		X acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afy	float	m/s/s		Y acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afz	float	m/s/s		Z acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
yaw	float	rad		yaw setpoint
yaw_rate	float	rad/s		yaw rate setpoint

POSITION_TARGET_GLOBAL_INT (#87

)

[Message] Reports the current commanded vehicle position, velocity, and acceleration as specified by the autopilot. This should match the commands sent in SET_POSITION_TARGET_GLOBAL_INT if the vehicle is being controlled this way.

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot). The rationale for the timestamp in the setpoint is to allow the system to compensate for the transport delay of the setpoint. This allows the system to compensate processing latency.
coordinate_frame	uint8_t		MAV_FRAME	Valid options are: MAV_FRAME_GLOBAL_INT = 5, MAV_FRAME_GLOBAL_RELATIVE_ALT_INT = 6, MAV_FRAME_GLOBAL_TERRAIN_ALT_INT = 11
type_mask	uint16_t		POSITION_TARGET_TYPEMASK	Bitmap to indicate which dimensions should be ignored by the vehicle.
lat_int	int32_t	degE7		X Position in WGS84 frame
lon_int	int32_t	degE7		Y Position in WGS84 frame
alt	float	m		Altitude (MSL, AGL or relative to home altitude, depending on frame)
vx	float	m/s		X velocity in NED frame
vy	float	m/s		Y velocity in NED frame
vz	float	m/s		Z velocity in NED frame
afx	float	m/s/s		X acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afy	float	m/s/s		Y acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afz	float	m/s/s		Z acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
yaw	float	rad		yaw setpoint
yaw_rate	float	rad/s		yaw rate setpoint

LOCAL_POSITION_NED_SYSTEM_GLOBAL_OFFSET (#89

)

[Message] The offset in X, Y, Z and yaw between the LOCAL_POSITION_NED messages of MAV X and the global coordinate frame in NED coordinates. Coordinate frame is right-handed, Z-axis down (aeronautical frame, NED / north-east-down convention)

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
x	float	m	X Position
y	float	m	Y Position
z	float	m	Z Position
roll	float	rad	Roll
pitch	float	rad	Pitch
yaw	float	rad	Yaw

HIL_STATE (#90

)

DEPRECATED: Replaced by HIL_STATE_QUATERNION (2013-07).

Suffers from missing airspeed fields and singularities due to Euler angles

[Message] Sent from simulation to autopilot. This packet is useful for high throughput applications such as hardware in the loop simulations.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
roll	float	rad	Roll angle
pitch	float	rad	Pitch angle
yaw	float	rad	Yaw angle
rollspeed	float	rad/s	Body frame roll / phi angular speed
pitchspeed	float	rad/s	Body frame pitch / theta angular speed
yawspeed	float	rad/s	Body frame yaw / psi angular speed
lat	int32_t	degE7	Latitude
lon	int32_t	degE7	Longitude
alt	int32_t	mm	Altitude
vx	int16_t	cm/s	Ground X Speed (Latitude)
vy	int16_t	cm/s	Ground Y Speed (Longitude)
vz	int16_t	cm/s	Ground Z Speed (Altitude)
xacc	int16_t	mG	X acceleration
yacc	int16_t	mG	Y acceleration
zacc	int16_t	mG	Z acceleration

HIL_CONTROLS (#91

)

[Message] Sent from autopilot to simulation. Hardware in the loop control outputs

Field Name	Type	Units	Values	Description
				Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer

time_usec	uint64_t	us	timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
roll_ailerons	float		Control output -1 .. 1
pitch_elevator	float		Control output -1 .. 1
yaw_rudder	float		Control output -1 .. 1
throttle	float		Throttle 0 .. 1
aux1	float		Aux 1, -1 .. 1
aux2	float		Aux 2, -1 .. 1
aux3	float		Aux 3, -1 .. 1
aux4	float		Aux 4, -1 .. 1
mode	uint8_t	MAV_MODE	System mode.
nav_mode	uint8_t		Navigation mode (MAV_NAV_MODE)

HIL_RC_INPUTS_RAW (#92

)

[Message] Sent from simulation to autopilot. The RAW values of the RC channels received. The standard PPM modulation is as follows: 1000 microseconds: 0%, 2000 microseconds: 100%. Individual receivers/transmitters might violate this specification.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
chan1_raw	uint16_t	us	RC channel 1 value
chan2_raw	uint16_t	us	RC channel 2 value
chan3_raw	uint16_t	us	RC channel 3 value
chan4_raw	uint16_t	us	RC channel 4 value
chan5_raw	uint16_t	us	RC channel 5 value
chan6_raw	uint16_t	us	RC channel 6 value
chan7_raw	uint16_t	us	RC channel 7 value
chan8_raw	uint16_t	us	RC channel 8 value
chan9_raw	uint16_t	us	RC channel 9 value
chan10_raw	uint16_t	us	RC channel 10 value
chan11_raw	uint16_t	us	RC channel 11 value
chan12_raw	uint16_t	us	RC channel 12 value
rssi	uint8_t		Receive signal strength indicator in device-dependent units/scale. Values: [0-254], UINT8_MAX: invalid/unknown.

HIL_ACTUATOR_CONTROLS (#93

)

[Message] Sent from autopilot to simulation. Hardware in the loop control outputs (replacement for HIL_CONTROLS)

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
controls	float[16]			Control outputs -1 .. 1. Channel assignment depends on the simulated hardware.
mode	uint8_t		MAV_MODE_FLAG	System mode. Includes arming state.
flags	uint64_t			Flags as bitfield, 1: indicate simulation using lockstep.

OPTICAL_FLOW (#100

)

[Message] Optical flow from a flow sensor (e.g. optical mouse sensor)

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
sensor_id	uint8_t		Sensor ID
flow_x	int16_t	dpix	Flow in x-sensor direction
flow_y	int16_t	dpix	Flow in y-sensor direction
flow_comp_m_x	float	m/s	Flow in x-sensor direction, angular-speed compensated
flow_comp_m_y	float	m/s	Flow in y-sensor direction, angular-speed compensated
quality	uint8_t		Optical flow quality / confidence. 0: bad, 255: maximum quality
ground_distance	float	m	Ground distance. Positive value: distance known. Negative value: Unknown distance
flow_rate_x**	float	rad/s	Flow rate about X axis
flow_rate_y**	float	rad/s	Flow rate about Y axis

GLOBAL_VISION_POSITION_ESTIMATE (#101

)

[Message] Global position/attitude estimate from a vision source.

Field Name	Type	Units	Description
usec	uint64_t	us	Timestamp (UNIX time or since system boot)
x	float	m	Global X position
y	float	m	Global Y position
z	float	m	Global Z position
roll	float	rad	Roll angle

pitch	float	rad	Pitch angle
yaw	float	rad	Yaw angle
covariance**	float[21]		Row-major representation of pose 6x6 cross-covariance matrix upper right triangle (states: x_global, y_global, z_global, roll, pitch, yaw; first six entries are the first ROW, next five entries are the second ROW, etc.). If unknown, assign NaN value to first element in the array.
reset_counter**	uint8_t		Estimate reset counter. This should be incremented when the estimate resets in any of the dimensions (position, velocity, attitude, angular speed). This is designed to be used when e.g an external SLAM system detects a loop-closure and the estimate jumps.

VISION_POSITION_ESTIMATE (#102

)

[Message] Local position/attitude estimate from a vision source.

Field Name	Type	Units	Description
usec	uint64_t	us	Timestamp (UNIX time or time since system boot)
x	float	m	Local X position
y	float	m	Local Y position
z	float	m	Local Z position
roll	float	rad	Roll angle
pitch	float	rad	Pitch angle
yaw	float	rad	Yaw angle
covariance**	float[21]		Row-major representation of pose 6x6 cross-covariance matrix upper right triangle (states: x, y, z, roll, pitch, yaw; first six entries are the first ROW, next five entries are the second ROW, etc.). If unknown, assign NaN value to first element in the array.
reset_counter**	uint8_t		Estimate reset counter. This should be incremented when the estimate resets in any of the dimensions (position, velocity, attitude, angular speed). This is designed to be used when e.g an external SLAM system detects a loop-closure and the estimate jumps.

VISION_SPEED_ESTIMATE (#103

)

[Message] Speed estimate from a vision source.

Field Name	Type	Units	Description
usec	uint64_t	us	Timestamp (UNIX time or time since system boot)
x	float	m/s	Global X speed
y	float	m/s	Global Y speed
z	float	m/s	Global Z speed
covariance**	float[9]		Row-major representation of 3x3 linear velocity covariance matrix (states: vx, vy, vz; 1st three entries - 1st row, etc.). If unknown, assign NaN value to first element in the array.
reset_counter**	uint8_t		Estimate reset counter. This should be incremented when the estimate resets in any of the dimensions (position, velocity, attitude, angular speed). This is designed to be used when e.g an external SLAM system detects a loop-closure and the estimate jumps.

VICON_POSITION_ESTIMATE (#104

)

[Message] Global position estimate from a Vicon motion system source.

Field Name	Type	Units	Description
usec	uint64_t	us	Timestamp (UNIX time or time since system boot)
x	float	m	Global X position
y	float	m	Global Y position
z	float	m	Global Z position
roll	float	rad	Roll angle
pitch	float	rad	Pitch angle
yaw	float	rad	Yaw angle
covariance**	float[21]		Row-major representation of 6x6 pose cross-covariance matrix upper right triangle (states: x, y, z, roll, pitch, yaw; first six entries are the first ROW, next five entries are the second ROW, etc.). If unknown, assign NaN value to first element in the array.

HIGHRES_IMU (#105

)

[Message] The IMU readings in SI units in NED body frame

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
xacc	float	m/s/s		X acceleration
yacc	float	m/s/s		Y acceleration
zacc	float	m/s/s		Z acceleration
xgyro	float	rad/s		Angular speed around X axis
ygyro	float	rad/s		Angular speed around Y axis
zgyro	float	rad/s		Angular speed around Z axis
xmag	float	gauss		X Magnetic field
ymag	float	gauss		Y Magnetic field
zmag	float	gauss		Z Magnetic field
abs_pressure	float	hPa		Absolute pressure
diff_pressure	float	hPa		Differential pressure

pressure_alt	float			Altitude calculated from pressure
temperature	float	degC		Temperature
fields_updated	uint16_t		HIGHRES_IMU_UPDATED_FLAGS	Bitmap for fields that have updated since last message
id**	uint8_t			Id. Ids are numbered from 0 and map to IMUs numbered from 1 (e.g. IMU1 will have a message with id=0)

OPTICAL_FLOW_RAD (#106

)

[Message] Optical flow from an angular rate flow sensor (e.g. PX4FLOW or mouse sensor)

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
sensor_id	uint8_t		Sensor ID
integration_time_us	uint32_t	us	Integration time. Divide integrated_x and integrated_y by the integration time to obtain average flow. The integration time also indicates the.
integrated_x	float	rad	Flow around X axis (Sensor RH rotation about the X axis induces a positive flow. Sensor linear motion along the positive Y axis induces a negative flow.)
integrated_y	float	rad	Flow around Y axis (Sensor RH rotation about the Y axis induces a positive flow. Sensor linear motion along the positive X axis induces a positive flow.)
integrated_xgyro	float	rad	RH rotation around X axis
integrated_ygyro	float	rad	RH rotation around Y axis
integrated_zgyro	float	rad	RH rotation around Z axis
temperature	int16_t	cdegC	Temperature
quality	uint8_t		Optical flow quality / confidence. 0: no valid flow, 255: maximum quality
time_delta_distance_us	uint32_t	us	Time since the distance was sampled.
distance	float	m	Distance to the center of the flow field. Positive value (including zero): distance known. Negative value: Unknown distance.

HIL_SENSOR (#107

)

[Message] The IMU readings in SI units in NED body frame

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
xacc	float	m/s/s		X acceleration
yacc	float	m/s/s		Y acceleration
zacc	float	m/s/s		Z acceleration
xgyro	float	rad/s		Angular speed around X axis in body frame
ygyro	float	rad/s		Angular speed around Y axis in body frame
zgyro	float	rad/s		Angular speed around Z axis in body frame
xmag	float	gauss		X Magnetic field
ymag	float	gauss		Y Magnetic field
zmag	float	gauss		Z Magnetic field
abs_pressure	float	hPa		Absolute pressure
diff_pressure	float	hPa		Differential pressure (airspeed)
pressure_alt	float			Altitude calculated from pressure
temperature	float	degC		Temperature
fields_updated	uint32_t		HIL_SENSOR_UPDATED_FLAGS	Bitmap for fields that have updated since last message
id**	uint8_t			Sensor ID (zero indexed). Used for multiple sensor inputs

SIM_STATE (#108

)

[Message] Status of simulation environment, if used

Field Name	Type	Units	Description
q1	float		True attitude quaternion component 1, w (1 in null-rotation)
q2	float		True attitude quaternion component 2, x (0 in null-rotation)
q3	float		True attitude quaternion component 3, y (0 in null-rotation)
q4	float		True attitude quaternion component 4, z (0 in null-rotation)
roll	float		Attitude roll expressed as Euler angles, not recommended except for human-readable outputs
pitch	float		Attitude pitch expressed as Euler angles, not recommended except for human-readable outputs
yaw	float		Attitude yaw expressed as Euler angles, not recommended except for human-readable outputs
xacc	float	m/s/s	X acceleration
yacc	float	m/s/s	Y acceleration
zacc	float	m/s/s	Z acceleration
xgyro	float	rad/s	Angular speed around X axis
ygyro	float	rad/s	Angular speed around Y axis
zgyro	float	rad/s	Angular speed around Z axis
lat	float	deg	Latitude
lon	float	deg	Longitude
alt	float	m	Altitude
std_dev_horz	float		Horizontal position standard deviation
std_dev_vert	float		Vertical position standard deviation
vn	float	m/s	True velocity in north direction in earth-fixed NED frame
ve	float	m/s	True velocity in east direction in earth-fixed NED frame

vd	float	m/s	True velocity in down direction in earth-fixed NED frame

RADIO_STATUS (#109

)

[Message] Status generated by radio and injected into MAVLink stream.

Field Name	Type	Units	Description
rssi	uint8_t		Local (message sender) received signal strength indication in device-dependent units/scale. Values: [0-254], UINT8_MAX: invalid/unknown.
remrssi	uint8_t		Remote (message receiver) signal strength indication in device-dependent units/scale. Values: [0-254], UINT8_MAX: invalid/unknown.
txbuf	uint8_t	%	Remaining free transmitter buffer space.
noise	uint8_t		Local background noise level. These are device dependent RSSI values (scale as approx 2x dB on SiK radios). Values: [0-254], UINT8_MAX: invalid/unknown.
remnoise	uint8_t		Remote background noise level. These are device dependent RSSI values (scale as approx 2x dB on SiK radios). Values: [0-254], UINT8_MAX: invalid/unknown.
rxerrors	uint16_t		Count of radio packet receive errors (since boot).
fixed	uint16_t		Count of error corrected radio packets (since boot).

FILE_TRANSFER_PROTOCOL (#110

)

[Message] File transfer protocol message: https://mavlink.io/en/services/ftp.html.

Field Name	Type	Description
target_network	uint8_t	Network ID (0 for broadcast)
target_system	uint8_t	System ID (0 for broadcast)
target_component	uint8_t	Component ID (0 for broadcast)
payload	uint8_t[251]	Variable length payload. The length is defined by the remaining message length when subtracting the header and other fields. The content/format of this block is defined in https://mavlink.io/en/services/ftp.html.

TIMESYNC (#111

)

[Message] Time synchronization message.

Field Name	Type	Description
tc1	int64_t	Time sync timestamp 1
ts1	int64_t	Time sync timestamp 2

CAMERA_TRIGGER (#112

)

[Message] Camera-IMU triggering and synchronisation message.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp for image frame (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
seq	uint32_t		Image frame sequence

HIL_GPS (#113

)

[Message] The global position, as returned by the Global Positioning System (GPS). This is NOT the global position estimate of the system, but rather a RAW sensor value. See message GLOBAL_POSITION_INT for the global position estimate.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
fix_type	uint8_t		0-1: no fix, 2: 2D fix, 3: 3D fix. Some applications will not use the value of this field unless it is at least two, so always correctly fill in the fix.
lat	int32_t	degE7	Latitude (WGS84)
lon	int32_t	degE7	Longitude (WGS84)
alt	int32_t	mm	Altitude (MSL). Positive for up.
eph	uint16_t		GPS HDOP horizontal dilution of position (unitless * 100). If unknown, set to: UINT16_MAX
epv	uint16_t		GPS VDOP vertical dilution of position (unitless * 100). If unknown, set to: UINT16_MAX
vel	uint16_t	cm/s	GPS ground speed. If unknown, set to: UINT16_MAX
vn	int16_t	cm/s	GPS velocity in north direction in earth-fixed NED frame
ve	int16_t	cm/s	GPS velocity in east direction in earth-fixed NED frame
vd	int16_t	cm/s	GPS velocity in down direction in earth-fixed NED frame
cog	uint16_t	cdeg	Course over ground (NOT heading, but direction of movement), 0.0..359.99 degrees. If unknown, set to: UINT16_MAX
satellites_visible	uint8_t		Number of satellites visible. If unknown, set to UINT8_MAX
id**	uint8_t		GPS ID (zero indexed). Used for multiple GPS inputs
yaw**	uint16_t	cdeg	Yaw of vehicle relative to Earth's North, zero means not available, use 36000 for north

HIL_OPTICAL_FLOW (#114

)

[Message] Simulated optical flow from a flow sensor (e.g. PX4FLOW or optical mouse sensor)

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
sensor_id	uint8_t		Sensor ID
integration_time_us	uint32_t	us	Integration time. Divide integrated_x and integrated_y by the integration time to obtain average flow. The integration time also indicates the.
integrated_x	float	rad	Flow in radians around X axis (Sensor RH rotation about the X axis induces a positive flow. Sensor linear motion along the positive Y axis induces a negative flow.)
integrated_y	float	rad	Flow in radians around Y axis (Sensor RH rotation about the Y axis induces a positive flow. Sensor linear motion along the positive X axis induces a positive flow.)
integrated_xgyro	float	rad	RH rotation around X axis
integrated_ygyro	float	rad	RH rotation around Y axis
integrated_zgyro	float	rad	RH rotation around Z axis
temperature	int16_t	cdegC	Temperature
quality	uint8_t		Optical flow quality / confidence. 0: no valid flow, 255: maximum quality
time_delta_distance_us	uint32_t	us	Time since the distance was sampled.
distance	float	m	Distance to the center of the flow field. Positive value (including zero): distance known. Negative value: Unknown distance.

HIL_STATE_QUATERNION (#115

)

[Message] Sent from simulation to autopilot, avoids in contrast to HIL_STATE singularities. This packet is useful for high throughput applications such as hardware in the loop simulations.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
attitude_quaternion	float[4]		Vehicle attitude expressed as normalized quaternion in w, x, y, z order (with 1 0 0 0 being the null-rotation)
rollspeed	float	rad/s	Body frame roll / phi angular speed
pitchspeed	float	rad/s	Body frame pitch / theta angular speed
yawspeed	float	rad/s	Body frame yaw / psi angular speed
lat	int32_t	degE7	Latitude
lon	int32_t	degE7	Longitude
alt	int32_t	mm	Altitude
vx	int16_t	cm/s	Ground X Speed (Latitude)
vy	int16_t	cm/s	Ground Y Speed (Longitude)
vz	int16_t	cm/s	Ground Z Speed (Altitude)
ind_airspeed	uint16_t	cm/s	Indicated airspeed
true_airspeed	uint16_t	cm/s	True airspeed
xacc	int16_t	mG	X acceleration
yacc	int16_t	mG	Y acceleration
zacc	int16_t	mG	Z acceleration

SCALED_IMU2 (#116

)

[Message] The RAW IMU readings for secondary 9DOF sensor setup. This message should contain the scaled values to the described units

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
xacc	int16_t	mG	X acceleration
yacc	int16_t	mG	Y acceleration
zacc	int16_t	mG	Z acceleration
xgyro	int16_t	mrad/s	Angular speed around X axis
ygyro	int16_t	mrad/s	Angular speed around Y axis
zgyro	int16_t	mrad/s	Angular speed around Z axis
xmag	int16_t	mgauss	X Magnetic field
ymag	int16_t	mgauss	Y Magnetic field
zmag	int16_t	mgauss	Z Magnetic field
temperature**	int16_t	cdegC	Temperature, 0: IMU does not provide temperature values. If the IMU is at 0C it must send 1 (0.01C).

LOG_REQUEST_LIST (#117

)

[Message] Request a list of available logs. On some systems calling this may stop on-board logging until LOG_REQUEST_END is called. If there are no log files available this request shall be answered with one LOG_ENTRY message with id = 0 and num_logs = 0.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
start	uint16_t	First log id (0 for first available)
end	uint16_t	Last log id (0xffff for last available)

LOG_ENTRY (#118

)

[Message] Reply to LOG_REQUEST_LIST

Field Name	Type	Units	Description
id	uint16_t		Log id
num_logs	uint16_t		Total number of logs
last_log_num	uint16_t		High log number
time_utc	uint32_t	s	UTC timestamp of log since 1970, or 0 if not available
size	uint32_t	bytes	Size of the log (may be approximate)

LOG_REQUEST_DATA (#119

)

[Message] Request a chunk of a log

Field Name	Type	Units	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
id	uint16_t		Log id (from LOG_ENTRY reply)
ofs	uint32_t		Offset into the log
count	uint32_t	bytes	Number of bytes

LOG_DATA (#120

)

[Message] Reply to LOG_REQUEST_DATA

Field Name	Type	Units	Description
id	uint16_t		Log id (from LOG_ENTRY reply)
ofs	uint32_t		Offset into the log
count	uint8_t	bytes	Number of bytes (zero for end of log)
data	uint8_t[90]		log data

LOG_ERASE (#121

)

[Message] Erase all logs

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID

LOG_REQUEST_END (#122

)

[Message] Stop log transfer and resume normal logging

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID

GPS_INJECT_DATA (#123

)

DEPRECATED: Replaced by GPS_RTCM_DATA (2022-05).

[Message] Data for injecting into the onboard GPS (used for DGPS)

Field Name	Type	Units	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
len	uint8_t	bytes	Data length
data	uint8_t[110]		Raw data (110 is enough for 12 satellites of RTCMv2)

GPS2_RAW (#124

)

[Message] Second GPS data.

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
fix_type	uint8_t		GPS_FIX_TYPE	GPS fix type.
lat	int32_t	degE7		Latitude (WGS84)
lon	int32_t	degE7		Longitude (WGS84)
alt	int32_t	mm		Altitude (MSL). Positive for up.
eph	uint16_t			GPS HDOP horizontal dilution of position (unitless * 100). If unknown, set to: UINT16_MAX

epv	uint16_t		GPS VDOP vertical dilution of position (unitless * 100). If unknown, set to: UINT16_MAX
vel	uint16_t	cm/s	GPS ground speed. If unknown, set to: UINT16_MAX
cog	uint16_t	cdeg	Course over ground (NOT heading, but direction of movement): 0.0..359.99 degrees. If unknown, set to: UINT16_MAX
satellites_visible	uint8_t		Number of satellites visible. If unknown, set to UINT8_MAX
dgps_numch	uint8_t		Number of DGPS satellites
dgps_age	uint32_t	ms	Age of DGPS info
yaw**	uint16_t	cdeg	Yaw in earth frame from north. Use 0 if this GPS does not provide yaw. Use UINT16_MAX if this GPS is configured to provide yaw and is currently unable to provide it. Use 36000 for north.
alt_ellipsoid**	int32_t	mm	Altitude (above WGS84, EGM96 ellipsoid). Positive for up.
h_acc**	uint32_t	mm	Position uncertainty.
v_acc**	uint32_t	mm	Altitude uncertainty.
vel_acc**	uint32_t	mm	Speed uncertainty.
hdg_acc**	uint32_t	degE5	Heading / track uncertainty

POWER_STATUS (#125

)

[Message] Power supply status

Field Name	Type	Units	Values	Description
Vcc	uint16_t	mV		5V rail voltage.
Vservo	uint16_t	mV		Servo rail voltage.
flags	uint16_t		MAV_POWER_STATUS	Bitmap of power supply status flags.

SERIAL_CONTROL (#126

)

[Message] Control a serial port. This can be used for raw access to an onboard serial peripheral such as a GPS or telemetry radio. It is designed to make it possible to update the devices firmware via MAVLink messages or change the devices settings. A message with zero bytes can be used to change just the baudrate.

Field Name	Type	Units	Values	Description
device	uint8_t		SERIAL_CONTROL_DEV	Serial control device type.
flags	uint8_t		SERIAL_CONTROL_FLAG	Bitmap of serial control flags.
timeout	uint16_t	ms		Timeout for reply data
baudrate	uint32_t	bits/s		Baudrate of transfer. Zero means no change.
count	uint8_t	bytes		how many bytes in this transfer
data	uint8_t[70]			serial data
target_system**	uint8_t			System ID
target_component**	uint8_t			Component ID

GPS_RTK (#127

)

[Message] RTK GPS data. Gives information on the relative baseline calculation the GPS is reporting

Field Name	Type	Units	Values	Description
time_last_baseline_ms	uint32_t	ms		Time since boot of last baseline message received.
rtk_receiver_id	uint8_t			Identification of connected RTK receiver.
wn	uint16_t			GPS Week Number of last baseline
tow	uint32_t	ms		GPS Time of Week of last baseline
rtk_health	uint8_t			GPS-specific health report for RTK data.
rtk_rate	uint8_t	Hz		Rate of baseline messages being received by GPS
nsats	uint8_t			Current number of sats used for RTK calculation.
baseline_coords_type	uint8_t		RTK_BASELINE_COORDINATE_SYSTEM	Coordinate system of baseline
baseline_a_mm	int32_t	mm		Current baseline in ECEF x or NED north component.
baseline_b_mm	int32_t	mm		Current baseline in ECEF y or NED east component.
baseline_c_mm	int32_t	mm		Current baseline in ECEF z or NED down component.
accuracy	uint32_t			Current estimate of baseline accuracy.
iar_num_hypotheses	int32_t			Current number of integer ambiguity hypotheses.

GPS2_RTK (#128

)

[Message] RTK GPS data. Gives information on the relative baseline calculation the GPS is reporting

Field Name	Type	Units	Values	Description
time_last_baseline_ms	uint32_t	ms		Time since boot of last baseline message received.
rtk_receiver_id	uint8_t			Identification of connected RTK receiver.
wn	uint16_t			GPS Week Number of last baseline
tow	uint32_t	ms		GPS Time of Week of last baseline
rtk_health	uint8_t			GPS-specific health report for RTK data.
rtk_rate	uint8_t	Hz		Rate of baseline messages being received by GPS
nsats	uint8_t			Current number of sats used for RTK calculation.
baseline_coords_type	uint8_t		RTK_BASELINE_COORDINATE_SYSTEM	Coordinate system of baseline
baseline_a_mm	int32_t	mm		Current baseline in ECEF x or NED north component.
baseline_b_mm	int32_t	mm		Current baseline in ECEF y or NED east component.
baseline_c_mm	int32_t	mm		Current baseline in ECEF z or NED down component.

accuracy	uint32_t	Current estimate of baseline accuracy.
iar_num_hypotheses	int32_t	Current number of integer ambiguity hypotheses.

SCALED_IMU3 (#129

)

[Message] The RAW IMU readings for 3rd 9DOF sensor setup. This message should contain the scaled values to the described units

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
xacc	int16_t	mG	X acceleration
yacc	int16_t	mG	Y acceleration
zacc	int16_t	mG	Z acceleration
xgyro	int16_t	mrad/s	Angular speed around X axis
ygyro	int16_t	mrad/s	Angular speed around Y axis
zgyro	int16_t	mrad/s	Angular speed around Z axis
xmag	int16_t	mgauss	X Magnetic field
ymag	int16_t	mgauss	Y Magnetic field
zmag	int16_t	mgauss	Z Magnetic field
temperature**	int16_t	cdegC	Temperature, 0: IMU does not provide temperature values. If the IMU is at 0C it must send 1 (0.01C).

DATA_TRANSMISSION_HANDSHAKE (#130

)

[Message] Handshake message to initiate, control and stop image streaming when using the Image Transmission Protocol: https://mavlink.io/en/services/image_transmission.html.

Field Name	Type	Units	Values	Description
type	uint8_t		MAVLINK_DATA_STREAM_TYPE	Type of requested/acknowledged data.
size	uint32_t	bytes		total data size (set on ACK only).
width	uint16_t			Width of a matrix or image.
height	uint16_t			Height of a matrix or image.
packets	uint16_t			Number of packets being sent (set on ACK only).
payload	uint8_t	bytes		Payload size per packet (normally 253 byte, see DATA field size in message ENCAPSULATED_DATA) (set on ACK only).
jpg_quality	uint8_t	%		JPEG quality. Values: [1-100].

ENCAPSULATED_DATA (#131

)

[Message] Data packet for images sent using the Image Transmission Protocol: https://mavlink.io/en/services/image_transmission.html.

Field Name	Type	Description
seqnr	uint16_t	sequence number (starting with 0 on every transmission)
data	uint8_t[253]	image data bytes

DISTANCE_SENSOR (#132

)

[Message] Distance sensor information for an onboard rangefinder.

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
min_distance	uint16_t	cm		Minimum distance the sensor can measure
max_distance	uint16_t	cm		Maximum distance the sensor can measure
current_distance	uint16_t	cm		Current distance reading
type	uint8_t		MAV_DISTANCE_SENSOR	Type of distance sensor.
id	uint8_t			Onboard ID of the sensor
orientation	uint8_t		MAV_SENSOR_ORIENTATION	Direction the sensor faces. downward-facing: ROTATION_PITCH_270, upward-facing: ROTATION_PITCH_90, backward-facing: ROTATION_PITCH_180, forward-facing: ROTATION_NONE, left-facing: ROTATION_YAW_90, right-facing: ROTATION_YAW_270
covariance	uint8_t	cm^2		Measurement variance. Max standard deviation is 6cm. UINT8_MAX if unknown.
horizontal_fov**	float	rad		Horizontal Field of View (angle) where the distance measurement is valid and the field of view is known. Otherwise this is set to 0.
vertical_fov**	float	rad		Vertical Field of View (angle) where the distance measurement is valid and the field of view is known. Otherwise this is set to 0.

quaternion**	float[4]		Quaternion of the sensor orientation in vehicle body frame (w, x, y, z order, zero-rotation is 1, 0, 0, 0). Zero-rotation is along the vehicle body x-axis. This field is required if the orientation is set to MAV_SENSOR_ROTATION_CUSTOM. Set it to 0 if invalid."
signal_quality**	uint8_t	%	Signal quality of the sensor. Specific to each sensor type, representing the relation of the signal strength with the target reflectivity, distance, size or aspect, but normalised as a percentage. 0 = unknown/unset signal quality, 1 = invalid signal, 100 = perfect signal.

TERRAIN_REQUEST (#133

)

[Message] Request for terrain data and terrain status. See terrain protocol docs: https://mavlink.io/en/services/terrain.html

Field Name	Type	Units	Description
lat	int32_t	degE7	Latitude of SW corner of first grid
lon	int32_t	degE7	Longitude of SW corner of first grid
grid_spacing	uint16_t	m	Grid spacing
mask	uint64_t		Bitmask of requested 4x4 grids (row major 8x7 array of grids, 56 bits)

TERRAIN_DATA (#134

)

[Message] Terrain data sent from GCS. The lat/lon and grid_spacing must be the same as a lat/lon from a TERRAIN_REQUEST. See terrain protocol docs: https://mavlink.io/en/services/terrain.html

Field Name	Type	Units	Description
lat	int32_t	degE7	Latitude of SW corner of first grid
lon	int32_t	degE7	Longitude of SW corner of first grid
grid_spacing	uint16_t	m	Grid spacing
gridbit	uint8_t		bit within the terrain request mask
data	int16_t[16]	m	Terrain data MSL

TERRAIN_CHECK (#135

)

[Message] Request that the vehicle report terrain height at the given location (expected response is a TERRAIN_REPORT). Used by GCS to check if vehicle has all terrain data needed for a mission.

Field Name	Type	Units	Description
lat	int32_t	degE7	Latitude
lon	int32_t	degE7	Longitude

TERRAIN_REPORT (#136

)

[Message] Streamed from drone to report progress of terrain map download (initiated by TERRAIN_REQUEST), or sent as a response to a TERRAIN_CHECK request. See terrain protocol docs: https://mavlink.io/en/services/terrain.html

Field Name	Type	Units	Description
lat	int32_t	degE7	Latitude
lon	int32_t	degE7	Longitude
spacing	uint16_t		grid spacing (zero if terrain at this location unavailable)
terrain_height	float	m	Terrain height MSL
current_height	float	m	Current vehicle height above lat/lon terrain height
pending	uint16_t		Number of 4x4 terrain blocks waiting to be received or read from disk
loaded	uint16_t		Number of 4x4 terrain blocks in memory

SCALED_PRESSURE2 (#137

)

[Message] Barometer readings for 2nd barometer

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
press_abs	float	hPa	Absolute pressure
press_diff	float	hPa	Differential pressure
temperature	int16_t	cdegC	Absolute pressure temperature
temperature_press_diff**	int16_t	cdegC	Differential pressure temperature (0, if not available). Report values of 0 (or 1) as 1 cdegC.

ATT_POS_MOCAP (#138

)

[Message] Motion capture attitude and position

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
q	float[4]		Attitude quaternion (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
x	float	m	X position (NED)
y	float	m	Y position (NED)
z	float	m	Z position (NED)
covariance**	float[21]		Row-major representation of a pose 6x6 cross-covariance matrix upper right triangle (states: x, y, z, roll, pitch, yaw; first six entries are the first ROW, next five entries are the second ROW, etc.). If unknown, assign NaN value to first element in the array.

SET_ACTUATOR_CONTROL_TARGET (#139

)

[Message] Set the vehicle attitude and body angular rates.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
group_mlx	uint8_t		Actuator group. The "_mlx" indicates this is a multi-instance message and a MAVLink parser should use this field to difference between instances.
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
controls	float[8]		Actuator controls. Normed to -1..+1 where 0 is neutral position. Throttle for single rotation direction motors is 0..1, negative range for reverse direction. Standard mapping for attitude controls (group 0): (index 0-7): roll, pitch, yaw, throttle, flaps, spoilers, airbrakes, landing gear. Load a pass-through mixer to repurpose them as generic outputs.

ACTUATOR_CONTROL_TARGET (#140

)

[Message] Set the vehicle attitude and body angular rates.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
group_mlx	uint8_t		Actuator group. The "_mlx" indicates this is a multi-instance message and a MAVLink parser should use this field to difference between instances.
controls	float[8]		Actuator controls. Normed to -1..+1 where 0 is neutral position. Throttle for single rotation direction motors is 0..1, negative range for reverse direction. Standard mapping for attitude controls (group 0): (index 0-7): roll, pitch, yaw, throttle, flaps, spoilers, airbrakes, landing gear. Load a pass-through mixer to repurpose them as generic outputs.

ALTITUDE (#141

)

[Message] The current system altitude.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
altitude_monotonic	float	m	This altitude measure is initialized on system boot and monotonic (it is never reset, but represents the local altitude change). The only guarantee on this field is that it will never be reset and is consistent within a flight. The recommended value for this field is the uncorrected barometric altitude at boot time. This altitude will also drift and vary between flights.
altitude_amsl	float	m	This altitude measure is strictly above mean sea level and might be non-monotonic (it might reset on events like GPS lock or when a new QNH value is set). It should be the altitude to which global altitude waypoints are compared to. Note that it is *not* the GPS altitude, however, most GPS modules already output MSL by default and not the WGS84 altitude.
altitude_local	float	m	This is the local altitude in the local coordinate frame. It is not the altitude above home, but in reference to the coordinate origin (0, 0, 0). It is up-positive.
altitude_relative	float	m	This is the altitude above the home position. It resets on each change of the current home position.
altitude_terrain	float	m	This is the altitude above terrain. It might be fed by a terrain database or an altimeter. Values smaller than -1000 should be interpreted as unknown.
bottom_clearance	float	m	This is not the altitude, but the clear space below the system according to the fused clearance estimate. It generally should max out at the maximum range of e.g. the laser altimeter. It is generally a moving target. A negative value indicates no measurement available.

RESOURCE_REQUEST (#142

)

[Message] The autopilot is requesting a resource (file, binary, other type of data)

Field Name	Type	Description
request_id	uint8_t	Request ID. This ID should be re-used when sending back URI contents
uri_type	uint8_t	The type of requested URI. 0 = a file via URL. 1 = a UAVCAN binary
uri	uint8_t[120]	The requested unique resource identifier (URI). It is not necessarily a straight domain name (depends on the URI type enum)
transfer_type	uint8_t	The way the autopilot wants to receive the URI. 0 = MAVLink FTP. 1 = binary stream.
storage	uint8_t[120]	The storage path the autopilot wants the URI to be stored in. Will only be valid if the transfer_type has a storage associated (e.g. MAVLink FTP).

SCALED_PRESSURE3 (#143

)

[Message] Barometer readings for 3rd barometer

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
press_abs	float	hPa	Absolute pressure
press_diff	float	hPa	Differential pressure
temperature	int16_t	cdegC	Absolute pressure temperature
temperature_press_diff**	int16_t	cdegC	Differential pressure temperature (0, if not available). Report values of 0 (or 1) as 1 cdegC.

FOLLOW_TARGET (#144

)

[Message] Current motion information from a designated system

Field Name	Type	Units	Description
timestamp	uint64_t	ms	Timestamp (time since system boot).
est_capabilities	uint8_t		bit positions for tracker reporting capabilities (POS = 0, VEL = 1, ACCEL = 2, ATT + RATES = 3)
lat	int32_t	degE7	Latitude (WGS84)
lon	int32_t	degE7	Longitude (WGS84)
alt	float	m	Altitude (MSL)
vel	float[3]	m/s	target velocity (0,0,0) for unknown
acc	float[3]	m/s/s	linear target acceleration (0,0,0) for unknown
attitude_q	float[4]		
(0 0 0 0 for unknown)			

| | rates | float[3] | | | (0 0 0 for unknown) | | position_cov | float[3] | | | eph epv | | custom_state | uint64_t | | button states or switches of a tracker device |

CONTROL_SYSTEM_STATE (#146

)

[Message] The smoothed, monotonic system state used to feed the control loops of the system.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
x_acc	float	m/s/s	X acceleration in body frame
y_acc	float	m/s/s	Y acceleration in body frame
z_acc	float	m/s/s	Z acceleration in body frame
x_vel	float	m/s	X velocity in body frame
y_vel	float	m/s	Y velocity in body frame
z_vel	float	m/s	Z velocity in body frame
x_pos	float	m	X position in local frame
y_pos	float	m	Y position in local frame
z_pos	float	m	Z position in local frame
airspeed	float	m/s	Airspeed, set to -1 if unknown
vel_variance	float[3]		Variance of body velocity estimate
pos_variance	float[3]		Variance in local position
q	float[4]		The attitude, represented as Quaternion
roll_rate	float	rad/s	Angular rate in roll axis
pitch_rate	float	rad/s	Angular rate in pitch axis
yaw_rate	float	rad/s	Angular rate in yaw axis

BATTERY_STATUS (#147

)

[Message] Battery information. Updates GCS with flight controller battery status. Smart batteries also use this message, but may additionally send SMART_BATTERY_INFO.

Field Name	Type	Units	Values	Description
id	uint8_t			Battery ID
battery_function	uint8_t		MAV_BATTERY_FUNCTION	Function of the battery
type	uint8_t		MAV_BATTERY_TYPE	Type (chemistry) of the battery
temperature	int16_t	cdegC		Temperature of the battery. INT16_MAX for unknown temperature.
voltages	uint16_t[10]	mV		Battery voltage of cells 1 to 10 (see voltages_ext for cells 11-14). Cells in this field above the valid cell count for this battery should have the UINT16_MAX value. If individual cell voltages are unknown or not measured for this battery, then the overall battery voltage should be filled in cell 0, with all others set to UINT16_MAX. If the voltage of the battery is greater than (UINT16_MAX - 1), then cell 0 should be set to (UINT16_MAX - 1), and cell 1 to the remaining voltage. This can be extended to multiple cells if the total voltage is greater than 2 * (UINT16_MAX - 1).
current_battery	int16_t	cA		Battery current, -1: autopilot does not measure the current
current_consumed	int32_t	mAh		Consumed charge, -1: autopilot does not provide consumption estimate
energy_consumed	int32_t	hJ		Consumed energy, -1: autopilot does not provide energy consumption estimate

battery_remaining	int8_t	%		Remaining battery energy. Values: [0-100], -1: autopilot does not estimate the remaining battery.
time_remaining**	int32_t	s		Remaining battery time, 0: autopilot does not provide remaining battery time estimate
charge_state**	uint8_t		MAV_BATTERY_CHARGE_STATE	State for extent of discharge, provided by autopilot for warning or external reactions
voltages_ext**	uint16_t[4]	mV		Battery voltages for cells 11 to 14. Cells above the valid cell count for this battery should have a value of 0, where zero indicates not supported (note, this is different than for the voltages field and allows empty byte truncation). If the measured value is 0 then 1 should be sent instead.
mode**	uint8_t		MAV_BATTERY_MODE	Battery mode. Default (0) is that battery mode reporting is not supported or battery is in normal-use mode.
fault_bitmask**	uint32_t		MAV_BATTERY_FAULT	Fault/health indications. These should be set when charge_state is MAV_BATTERY_CHARGE_STATE_FAILED or MAV_BATTERY_CHARGE_STATE_UNHEALTHY (if not, fault reporting is not supported).

AUTOPILOT_VERSION (#148

)

[Message] Version and capability of autopilot software. This should be emitted in response to a request with MAV_CMD_REQUEST_MESSAGE.

Field Name	Type	Values	Description
capabilities	uint64_t	MAV_PROTOCOL_CAPABILITY	Bitmap of capabilities
flight_sw_version	uint32_t		Firmware version number
middleware_sw_version	uint32_t		Middleware version number
os_sw_version	uint32_t		Operating system version number
board_version	uint32_t		HW / board version (last 8 bits should be silicon ID, if any). The first 16 bits of this field specify https://github.com/PX4/PX4-Bootloader/blob/master/board_types.txt
flight_custom_version	uint8_t[8]		Custom version field, commonly the first 8 bytes of the git hash. This is not an unique identifier, but should allow to identify the commit using the main version number even for very large code bases.
middleware_custom_version	uint8_t[8]		Custom version field, commonly the first 8 bytes of the git hash. This is not an unique identifier, but should allow to identify the commit using the main version number even for very large code bases.
os_custom_version	uint8_t[8]		Custom version field, commonly the first 8 bytes of the git hash. This is not an unique identifier, but should allow to identify the commit using the main version number even for very large code bases.
vendor_id	uint16_t		ID of the board vendor
product_id	uint16_t		ID of the product
uid	uint64_t		UID if provided by hardware (see uid2)
uid2**	uint8_t[18]		UID if provided by hardware (supersedes the uid field. If this is non-zero, use this field, otherwise use uid)

LANDING_TARGET (#149

)

[Message] The location of a landing target. See: https://mavlink.io/en/services/landing_target.html

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
target_num	uint8_t			The ID of the target if multiple targets are present
frame	uint8_t		MAV_FRAME	Coordinate frame used for following fields.
angle_x	float	rad		X-axis angular offset of the target from the center of the image
angle_y	float	rad		Y-axis angular offset of the target from the center of the image
distance	float	m		Distance to the target from the vehicle
size_x	float	rad		Size of target along x-axis
size_y	float	rad		Size of target along y-axis
x**	float	m		X Position of the landing target in MAV_FRAME
y**	float	m		Y Position of the landing target in MAV_FRAME
z**	float	m		Z Position of the landing target in MAV_FRAME
q**	float[4]			Quaternion of landing target orientation (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
type**	uint8_t		LANDING_TARGET_TYPE	Type of landing target
position_valid**	uint8_t			Boolean indicating whether the position fields (x, y, z, q, type) contain valid target position information (valid: 1, invalid: 0). Default is 0 (invalid).

FENCE_STATUS (#162

)

[Message] Status of geo-fencing. Sent in extended status stream when fencing enabled.

Field Name	Type	Units	Values	Description
breach_status	uint8_t			Breach status (0 if currently inside fence, 1 if outside).
breach_count	uint16_t			Number of fence breaches.
breach_type	uint8_t		FENCE_BREACH	Last breach type.
breach_time	uint32_t	ms		Time (since boot) of last breach.
breach_mitigation**	uint8_t		FENCE_MITIGATE	Active action to prevent fence breach

MAG_CAL_REPORT (#192

)

[Message] Reports results of completed compass calibration. Sent until MAG_CAL_ACK received.

Field Name	Type	Units	Values	Description
compass_id	uint8_t			Compass being calibrated.
cal_mask	uint8_t			Bitmask of compasses being calibrated.
cal_status	uint8_t		MAG_CAL_STATUS	Calibration Status.
autosaved	uint8_t			0=requires a MAV_CMD_DO_ACCEPT_MAG_CAL, 1=saved to parameters.
fitness	float	mgauss		RMS milligauss residuals.
ofs_x	float			X offset.
ofs_y	float			Y offset.
ofs_z	float			Z offset.
diag_x	float			X diagonal (matrix 11).
diag_y	float			Y diagonal (matrix 22).
diag_z	float			Z diagonal (matrix 33).
offdiag_x	float			X off-diagonal (matrix 12 and 21).
offdiag_y	float			Y off-diagonal (matrix 13 and 31).
offdiag_z	float			Z off-diagonal (matrix 32 and 23).
orientation_confidence**	float			Confidence in orientation (higher is better).
old_orientation**	uint8_t		MAV_SENSOR_ORIENTATION	orientation before calibration.
new_orientation**	uint8_t		MAV_SENSOR_ORIENTATION	orientation after calibration.
scale_factor**	float			field radius correction factor

EFI_STATUS (#225

)

[Message] EFI status output

Field Name	Type	Units	Description
health	uint8_t		EFI health status
ecu_index	float		ECU index
rpm	float		RPM
fuel_consumed	float	cm^3	Fuel consumed
fuel_flow	float	cm^3/min	Fuel flow rate
engine_load	float		
%			
Engine load			
throttle_position	float		
%			
Throttle position			
spark_dwell_time	float	ms	Spark dwell time
barometric_pressure	float	kPa	Barometric pressure
intake_manifold_pressure	float	kPa	Intake manifold pressure(
intake_manifold_temperature	float	degC	Intake manifold temperature
cylinder_head_temperature	float	degC	Cylinder head temperature
ignition_timing	float	deg	Ignition timing (Crank angle degrees)
injection_time	float	ms	Injection time
exhaust_gas_temperature	float	degC	Exhaust gas temperature
throttle_out	float		
%			
Output throttle			
pt_compensation	float		Pressure/temperature compensation
ignition_voltage**	float	V	Supply voltage to EFI sparking system. Zero in this value means "unknown", so if the supply voltage really is zero volts use 0.0001 instead.

ESTIMATOR_STATUS (#230

)

[Message] Estimator status message including flags, innovation test ratios and estimated accuracies. The flags message is an integer bitmask containing information on which EKF outputs are valid. See the ESTIMATOR_STATUS_FLAGS enum definition for further information. The innovation test ratios show the magnitude of the sensor innovation divided by the innovation check threshold. Under normal operation the innovation test ratios should be below 0.5 with occasional values up to 1.0. Values greater than 1.0 should be rare under normal operation and indicate that a measurement has been rejected by the filter. The user should be notified if an innovation test ratio greater than 1.0 is recorded. Notifications for values in the range between 0.5 and 1.0 should be optional and controllable by the user.

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.

flags	uint16_t		ESTIMATOR_STATUS_FLAGS	Bitmap indicating which EKF outputs are valid.
vel_ratio	float			Velocity innovation test ratio
pos_horiz_ratio	float			Horizontal position innovation test ratio
pos_vert_ratio	float			Vertical position innovation test ratio
mag_ratio	float			Magnetometer innovation test ratio
hagl_ratio	float			Height above terrain innovation test ratio
tas_ratio	float			True airspeed innovation test ratio
pos_horiz_accuracy	float	m		Horizontal position 1-STD accuracy relative to the EKF local origin
pos_vert_accuracy	float	m		Vertical position 1-STD accuracy relative to the EKF local origin

WIND_COV (#231

)

[Message] Wind estimate from vehicle. Note that despite the name, this message does not actually contain any covariances but instead variability and accuracy fields in terms of standard deviation (1-STD).

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
wind_x	float	m/s	Wind in North (NED) direction (NAN if unknown)
wind_y	float	m/s	Wind in East (NED) direction (NAN if unknown)
wind_z	float	m/s	Wind in down (NED) direction (NAN if unknown)
var_horiz	float	m/s	Variability of wind in XY, 1-STD estimated from a 1 Hz lowpassed wind estimate (NAN if unknown)
var_vert	float	m/s	Variability of wind in Z, 1-STD estimated from a 1 Hz lowpassed wind estimate (NAN if unknown)
wind_alt	float	m	Altitude (MSL) that this measurement was taken at (NAN if unknown)
horiz_accuracy	float	m/s	Horizontal speed 1-STD accuracy (0 if unknown)
vert_accuracy	float	m/s	Vertical speed 1-STD accuracy (0 if unknown)

GPS_INPUT (#232

)

[Message] GPS sensor input message. This is a raw sensor value sent by the GPS. This is NOT the global position estimate of the system.

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
gps_id	uint8_t			ID of the GPS for multiple GPS inputs
ignore_flags	uint16_t		GPS_INPUT_IGNORE_FLAGS	Bitmap indicating which GPS input flags fields to ignore. All other fields must be provided.
time_week_ms	uint32_t	ms		GPS time (from start of GPS week)
time_week	uint16_t			GPS week number
fix_type	uint8_t			0-1: no fix, 2: 2D fix, 3: 3D fix. 4: 3D with DGPS. 5: 3D with RTK
lat	int32_t	degE7		Latitude (WGS84)
lon	int32_t	degE7		Longitude (WGS84)
alt	float	m		Altitude (MSL). Positive for up.
hdop	float			GPS HDOP horizontal dilution of position (unitless). If unknown, set to: UINT16_MAX
vdop	float			GPS VDOP vertical dilution of position (unitless). If unknown, set to: UINT16_MAX
vn	float	m/s		GPS velocity in north direction in earth-fixed NED frame
ve	float	m/s		GPS velocity in east direction in earth-fixed NED frame
vd	float	m/s		GPS velocity in down direction in earth-fixed NED frame
speed_accuracy	float	m/s		GPS speed accuracy
horiz_accuracy	float	m		GPS horizontal accuracy
vert_accuracy	float	m		GPS vertical accuracy
satellites_visible	uint8_t			Number of satellites visible.
yaw**	uint16_t	cdeg		Yaw of vehicle relative to Earth's North, zero means not available, use 36000 for north

GPS_RTCM_DATA (#233

)

[Message] RTCM message for injecting into the onboard GPS (used for DGPS)

Field Name	Type	Units	Description
flags	uint8_t		LSB: 1 means message is fragmented, next 2 bits are the fragment ID, the remaining 5 bits are used for the sequence ID. Messages are only to be flushed to the GPS when the entire message has been reconstructed on the autopilot. The fragment ID specifies which order the fragments should be assembled into a buffer, while the sequence ID is used to detect a mismatch between different buffers. The buffer is considered fully reconstructed when either all 4 fragments are present, or all the fragments before the first fragment with a non full payload is received. This management is used to ensure that normal GPS operation doesn't corrupt RTCM data, and to recover from a unreliable transport delivery order.
len	uint8_t	bytes	data length
data	uint8_t[180]		RTCM message (may be fragmented)

HIGH_LATENCY (#234

)

DEPRECATED: Replaced by HIGH_LATENCY2 (2020-10).

[Message] Message appropriate for high latency connections like Iridium

Field Name	Type	Units	Values	Description
base_mode	uint8_t		MAV_MODE_FLAG	Bitmap of enabled system modes.
custom_mode	uint32_t			A bitfield for use for autopilot-specific flags.
landed_state	uint8_t		MAV_LANDED_STATE	The landed state. Is set to MAV_LANDED_STATE_UNDEFINED if landed state is unknown.
roll	int16_t	cdeg		roll
pitch	int16_t	cdeg		pitch
heading	uint16_t	cdeg		heading
throttle	int8_t	%		throttle (percentage)
heading_sp	int16_t	cdeg		heading setpoint
latitude	int32_t	degE7		Latitude
longitude	int32_t	degE7		Longitude
altitude_amsl	int16_t	m		Altitude above mean sea level
altitude_sp	int16_t	m		Altitude setpoint relative to the home position
airspeed	uint8_t	m/s		airspeed
airspeed_sp	uint8_t	m/s		airspeed setpoint
groundspeed	uint8_t	m/s		groundspeed
climb_rate	int8_t	m/s		climb rate
gps_nsat	uint8_t			Number of satellites visible. If unknown, set to UINT8_MAX
gps_fix_type	uint8_t		GPS_FIX_TYPE	GPS Fix type.
battery_remaining	uint8_t	%		Remaining battery (percentage)
temperature	int8_t	degC		Autopilot temperature (degrees C)
temperature_air	int8_t	degC		Air temperature (degrees C) from airspeed sensor
failsafe	uint8_t			failsafe (each bit represents a failsafe where 0=ok, 1=failsafe active (bit0:RC, bit1:batt, bit2:GPS, bit3:GCS, bit4:fence)
wp_num	uint8_t			current waypoint number
wp_distance	uint16_t	m		distance to target

HIGH_LATENCY2 (#235

)

[Message] Message appropriate for high latency connections like Iridium (version 2)

Field Name	Type	Units	Values	Description
timestamp	uint32_t	ms		Timestamp (milliseconds since boot or Unix epoch)
type	uint8_t		MAV_TYPE	Type of the MAV (quadrotor, helicopter, etc.)
autopilot	uint8_t		MAV_AUTOPILOT	Autopilot type / class. Use MAV_AUTOPILOT_INVALID for components that are not flight controllers.
custom_mode	uint16_t			A bitfield for use for autopilot-specific flags (2 byte version).
latitude	int32_t	degE7		Latitude
longitude	int32_t	degE7		Longitude
altitude	int16_t	m		Altitude above mean sea level
target_altitude	int16_t	m		Altitude setpoint
heading	uint8_t	deg/2		Heading
target_heading	uint8_t	deg/2		Heading setpoint
target_distance	uint16_t	dam		Distance to target waypoint or position
throttle	uint8_t	%		Throttle
airspeed	uint8_t	m/s*5		Airspeed
airspeed_sp	uint8_t	m/s*5		Airspeed setpoint
groundspeed	uint8_t	m/s*5		Groundspeed
windspeed	uint8_t	m/s*5		Windspeed
wind_heading	uint8_t	deg/2		Wind heading
eph	uint8_t	dm		Maximum error horizontal position since last message
epv	uint8_t	dm		Maximum error vertical position since last message
temperature_air	int8_t	degC		Air temperature from airspeed sensor
climb_rate	int8_t	dm/s		Maximum climb rate magnitude since last message
battery	int8_t	%		Battery level (-1 if field not provided).
wp_num	uint16_t			Current waypoint number
failure_flags	uint16_t		HL_FAILURE_FLAG	Bitmap of failure flags.
custom0	int8_t			Field for custom payload.
custom1	int8_t			Field for custom payload.
custom2	int8_t			Field for custom payload.

VIBRATION (#241

)

[Message] Vibration levels and accelerometer clipping

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
vibration_x	float		Vibration levels on X-axis
vibration_y	float		Vibration levels on Y-axis
vibration_z	float		Vibration levels on Z-axis
clipping_0	uint32_t		first accelerometer clipping count
clipping_1	uint32_t		second accelerometer clipping count
clipping_2	uint32_t		third accelerometer clipping count

HOME_POSITION (#242

)

[Message] Contains the home position. The home position is the default position that the system will return to and land on. The position must be set automatically by the system during the takeoff, and may also be explicitly set using MAV_CMD_DO_SET_HOME. The global and local positions encode the position in the respective coordinate frames, while the q parameter encodes the orientation of the surface. Under normal conditions it describes the heading and terrain slope, which can be used by the aircraft to adjust the approach. The approach 3D vector describes the point to which the system should fly in normal flight mode and then perform a landing sequence along the vector. Note: this message can be requested by sending the MAV_CMD_REQUEST_MESSAGE with param1=242 (or the deprecated MAV_CMD_GET_HOME_POSITION command).

Field Name	Type	Units	Description
latitude	int32_t	degE7	Latitude (WGS84)
longitude	int32_t	degE7	Longitude (WGS84)
altitude	int32_t	mm	Altitude (MSL). Positive for up.
x	float	m	Local X position of this position in the local coordinate frame
y	float	m	Local Y position of this position in the local coordinate frame
z	float	m	Local Z position of this position in the local coordinate frame
q	float[4]		World to surface normal and heading transformation of the takeoff position. Used to indicate the heading and slope of the ground
approach_x	float	m	Local X position of the end of the approach vector. Multicopters should set this position based on their takeoff path. Grass-landing fixed wing aircraft should set it the same way as multicopters. Runway-landing fixed wing aircraft should set it to the opposite direction of the takeoff, assuming the takeoff happened from the threshold / touchdown zone.
approach_y	float	m	Local Y position of the end of the approach vector. Multicopters should set this position based on their takeoff path. Grass-landing fixed wing aircraft should set it the same way as multicopters. Runway-landing fixed wing aircraft should set it to the opposite direction of the takeoff, assuming the takeoff happened from the threshold / touchdown zone.
approach_z	float	m	Local Z position of the end of the approach vector. Multicopters should set this position based on their takeoff path. Grass-landing fixed wing aircraft should set it the same way as multicopters. Runway-landing fixed wing aircraft should set it to the opposite direction of the takeoff, assuming the takeoff happened from the threshold / touchdown zone.
time_usec**	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.

SET_HOME_POSITION (#243

)

DEPRECATED: Replaced by MAV_CMD_DO_SET_HOME (2022-02).

The command protocol version (MAV_CMD_DO_SET_HOME) allows a GCS to detect when setting the home position has failed.

[Message] Sets the home position. The home position is the default position that the system will return to and land on. The position is set automatically by the system during the takeoff (and may also be set using this message). The global and local positions encode the position in the respective coordinate frames, while the q parameter encodes the orientation of the surface. Under normal conditions it describes the heading and terrain slope, which can be used by the aircraft to adjust the approach. The approach 3D vector describes the point to which the system should fly in normal flight mode and then perform a landing sequence along the vector. Note: the current home position may be emitted in a HOME_POSITION message on request (using MAV_CMD_REQUEST_MESSAGE with param1=242).

Field Name	Type	Units	Description
target_system	uint8_t		System ID.
latitude	int32_t	degE7	Latitude (WGS84)
longitude	int32_t	degE7	Longitude (WGS84)
altitude	int32_t	mm	Altitude (MSL). Positive for up.
x	float	m	Local X position of this position in the local coordinate frame
y	float	m	Local Y position of this position in the local coordinate frame
z	float	m	Local Z position of this position in the local coordinate frame
q	float[4]		World to surface normal and heading transformation of the takeoff position. Used to indicate the heading and slope of the ground
approach_x	float	m	Local X position of the end of the approach vector. Multicopters should set this position based on their takeoff path. Grass-landing fixed wing aircraft should set it the same way as multicopters. Runway-landing fixed wing aircraft should set it to the opposite direction of the takeoff, assuming the takeoff happened from the threshold / touchdown zone.
approach_y	float	m	Local Y position of the end of the approach vector. Multicopters should set this position based on their takeoff path. Grass-landing fixed wing aircraft should set it the same way as multicopters. Runway-landing fixed wing aircraft should set it to the opposite direction of the takeoff, assuming the takeoff happened from the threshold / touchdown zone.
approach_z	float	m	Local Z position of the end of the approach vector. Multicopters should set this position based on their takeoff path. Grass-landing fixed wing aircraft should set it the same way as multicopters. Runway-landing fixed wing aircraft should set it to the opposite direction of the takeoff, assuming the takeoff happened from the threshold / touchdown zone.
time_usec**	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.

MESSAGE_INTERVAL (#244

)

[Message] The interval between messages for a particular MAVLink message ID. This message is sent in response to the MAV_CMD_REQUEST_MESSAGE command with param1=244 (this message) and param2=message_id (the id of the message for which the interval is required). It may also be sent in response to MAV_CMD_GET_MESSAGE_INTERVAL. This interface replaces DATA_STREAM.

Field Name	Type	Units	Description
message_id	uint16_t		The ID of the requested MAVLink message. v1.0 is limited to 254 messages.
interval_us	int32_t	us	The interval between two messages. A value of -1 indicates this stream is disabled, 0 indicates it is not available, > 0 indicates the interval at which it is sent.

EXTENDED_SYS_STATE (#245

)

[Message] Provides state for additional features

Field Name	Type	Values	Description
vtol_state	uint8_t	MAV_VTOL_STATE	The VTOL state if applicable. Is set to MAV_VTOL_STATE_UNDEFINED if UAV is not in VTOL configuration.
landed_state	uint8_t	MAV_LANDED_STATE	The landed state. Is set to MAV_LANDED_STATE_UNDEFINED if landed state is unknown.

ADSB_VEHICLE (#246

)

[Message] The location and information of an ADSB vehicle

Field Name	Type	Units	Values	Description
ICAO_address	uint32_t			ICAO address
lat	int32_t	degE7		Latitude
lon	int32_t	degE7		Longitude
altitude_type	uint8_t		ADSB_ALTITUDE_TYPE	ADSB altitude type.
altitude	int32_t	mm		Altitude(ASL)
heading	uint16_t	cdeg		Course over ground
hor_velocity	uint16_t	cm/s		The horizontal velocity
ver_velocity	int16_t	cm/s		The vertical velocity. Positive is up
callsign	char[9]			The callsign, 8+null
emitter_type	uint8_t		ADSB_EMITTER_TYPE	ADSB emitter type.
tslc	uint8_t	s		Time since last communication in seconds
flags	uint16_t		ADSB_FLAGS	Bitmap to indicate various statuses including valid data fields
squawk	uint16_t			Squawk code

COLLISION (#247

)

[Message] Information about a potential collision

Field Name	Type	Units	Values	Description
src	uint8_t		MAV_COLLISION_SRC	Collision data source
id	uint32_t			Unique identifier, domain based on src field
action	uint8_t		MAV_COLLISION_ACTION	Action that is being taken to avoid this collision
threat_level	uint8_t		MAV_COLLISION_THREAT_LEVEL	How concerned the aircraft is about this collision
time_to_minimum_delta	float	s		Estimated time until collision occurs
altitude_minimum_delta	float	m		Closest vertical distance between vehicle and object
horizontal_minimum_delta	float	m		Closest horizontal distance between vehicle and object

V2_EXTENSION (#248

)

[Message] Message implementing parts of the V2 payload specs in V1 frames for transitional support.

Field Name	Type	Description
target_network	uint8_t	Network ID (0 for broadcast)
target_system	uint8_t	System ID (0 for broadcast)
target_component	uint8_t	Component ID (0 for broadcast)
message_type	uint16_t	A code that identifies the software component that understands this message (analogous to USB device classes or mime type strings). If this code is less than 32768, it is considered a 'registered' protocol extension and the corresponding entry should be added to https://github.com/mavlink/mavlink/definition_files/extension_message_ids.xml. Software creators can register blocks of message IDs as needed (useful for GCS specific metadata, etc...). Message_types greater than 32767 are considered local experiments and should not be checked in to any widely distributed codebase.
payload	uint8_t[249]	Variable length payload. The length must be encoded in the payload as part of the message_type protocol, e.g. by including the length as payload data, or by terminating the payload data with a non-zero marker. This is required in order to reconstruct zero-terminated payloads that are (or otherwise would be) trimmed by MAVLink 2 empty-byte truncation. The entire content of the payload block is opaque unless you understand the encoding message_type. The particular encoding used can be extension specific and might not always be documented as part of the MAVLink specification.

MEMORY_VECT (#249

)

[Message] Send raw controller memory. The use of this message is discouraged for normal packets, but a quite efficient way for testing new messages and getting experimental debug output.

Field Name	Type	Description
address	uint16_t	Starting address of the debug variables
ver	uint8_t	Version code of the type variable. 0=unknown, type ignored and assumed int16_t. 1=as below
type	uint8_t	Type code of the memory variables. for ver = 1: 0=16 x int16_t, 1=16 x uint16_t, 2=16 x Q15, 3=16 x 1Q14
value	int8_t[32]	Memory contents at specified address

DEBUG_VECT (#250

)

[Message] To debug something using a named 3D vector.

Field Name	Type	Units	Description
name	char[10]		Name
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
x	float		x
y	float		y
z	float		z

NAMED_VALUE_FLOAT (#251

)

[Message] Send a key-value pair as float. The use of this message is discouraged for normal packets, but a quite efficient way for testing new messages and getting experimental debug output.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
name	char[10]		Name of the debug variable
value	float		Floating point value

NAMED_VALUE_INT (#252

)

[Message] Send a key-value pair as integer. The use of this message is discouraged for normal packets, but a quite efficient way for testing new messages and getting experimental debug output.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
name	char[10]		Name of the debug variable
value	int32_t		Signed integer value

STATUSTEXT (#253

)

[Message] Status text message. These messages are printed in yellow in the COMM console of QGroundControl. WARNING: They consume quite some bandwidth, so use only for important status and error messages. If implemented wisely, these messages are buffered on the MCU and sent only at a limited rate (e.g. 10 Hz).

Field Name	Type	Values	Description
severity	uint8_t	MAV_SEVERITY	Severity of status. Relies on the definitions within RFC-5424.
text	char[50]		Status text message, without null termination character
id**	uint16_t		Unique (opaque) identifier for this statustext message. May be used to reassemble a logical long-statustext message from a sequence of chunks. A value of zero indicates this is the only chunk in the sequence and the message can be emitted immediately.
chunk_seq**	uint8_t		This chunk's sequence number; indexing is from zero. Any null character in the text field is taken to mean this was the last chunk.

DEBUG (#254

)

[Message] Send a debug value. The index is used to discriminate between values. These values show up in the plot of QGroundControl as DEBUG N.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
ind	uint8_t		index of debug variable
value	float		DEBUG value

SETUP_SIGNING (#256

)

[Message] **(MAVLink 2)** Setup a MAVLink2 signing key. If called with secret_key of all zero and zero initial_timestamp will disable signing

Field Name	Type	Description
target_system	uint8_t	system id of the target
target_component	uint8_t	component ID of the target
secret_key	uint8_t[32]	signing key

initial_timestamp	uint64_t	initial timestamp

BUTTON_CHANGE (#257

)

[Message] **(MAVLink 2)** Report button state change.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
last_change_ms	uint32_t	ms	Time of last change of button state.
state	uint8_t		Bitmap for state of buttons.

PLAY_TUNE (#258

)

DEPRECATED: Replaced by PLAY_TUNE_V2 (2019-10).

New version explicitly defines format. More interoperable.

[Message] **(MAVLink 2)** Control vehicle tone generation (buzzer).

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
tune	char[30]	tune in board specific format
tune2**	char[200]	tune extension (appended to tune)

CAMERA_INFORMATION (#259

)

[Message] **(MAVLink 2)** Information about a camera. Can be requested with a MAV_CMD_REQUEST_MESSAGE command.

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
vendor_name	uint8_t[32]			Name of the camera vendor
model_name	uint8_t[32]			Name of the camera model
firmware_version	uint32_t			Version of the camera firmware, encoded as: (Dev & 0xff) << 24
focal_length	float	mm		Focal length
sensor_size_h	float	mm		Image sensor size horizontal
sensor_size_v	float	mm		Image sensor size vertical
resolution_h	uint16_t	pix		Horizontal image resolution
resolution_v	uint16_t	pix		Vertical image resolution
lens_id	uint8_t			Reserved for a lens ID
flags	uint32_t		CAMERA_CAP_FLAGS	Bitmap of camera capability flags.
cam_definition_version	uint16_t			Camera definition version (iteration)
cam_definition_uri	char[140]			Camera definition URI (if any, otherwise only basic functions will be available). HTTP- (http://) and MAVLink FTP- (mavlinkftp://) formatted URIs are allowed (and both must be supported by any GCS that implements the Camera Protocol). The definition file may be xz compressed, which will be indicated by the file extension .xml.xz (a GCS that implements the protocol must support decompressing the file). The string needs to be zero terminated.

CAMERA_SETTINGS (#260

)

[Message] **(MAVLink 2)** Settings of a camera. Can be requested with a MAV_CMD_REQUEST_MESSAGE command.

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
mode_id	uint8_t		CAMERA_MODE	Camera mode
zoomLevel**	float			Current zoom level (0.0 to 100.0, NaN if not known)
focusLevel**	float			Current focus level (0.0 to 100.0, NaN if not known)

STORAGE_INFORMATION (#261

)

[Message] **(MAVLink 2)** Information about a storage medium. This message is sent in response to a request with MAV_CMD_REQUEST_MESSAGE and whenever the status of the storage changes (STORAGE_STATUS). Use MAV_CMD_REQUEST_MESSAGE.param2 to indicate the index/id of requested storage: 0 for all, 1 for first, 2 for second, etc.

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
storage_id	uint8_t			Storage ID (1 for first, 2 for second, etc.)
storage_count	uint8_t			Number of storage devices
status	uint8_t		STORAGE_STATUS	Status of storage
total_capacity	float	MiB		Total capacity. If storage is not ready (STORAGE_STATUS_READY) value will be ignored.
used_capacity	float	MiB		Used capacity. If storage is not ready (STORAGE_STATUS_READY) value will be ignored.

available_capacity	float	MiB		Available storage capacity. If storage is not ready (STORAGE_STATUS_READY) value will be ignored.
read_speed	float	MiB/s		Read speed.
write_speed	float	MiB/s		Write speed.
type**	uint8_t		STORAGE_TYPE	Type of storage
name**	char[32]			Textual storage name to be used in UI (microSD 1, Internal Memory, etc.) This is a NULL terminated string. If it is exactly 32 characters long, add a terminating NULL. If this string is empty, the generic type is shown to the user.
storage_usage**	uint8_t		STORAGE_USAGE_FLAG	Flags indicating whether this instance is preferred storage for photos, videos, etc. Note: Implementations should initially set the flags on the system-default storage id used for saving media (if possible/supported). This setting can then be overridden using MAV_CMD_SET_STORAGE_USAGE. If the media usage flags are not set, a GCS may assume storage ID 1 is the default storage for all media types.

CAMERA_CAPTURE_STATUS (#262

)

[Message] **(MAVLink 2)** Information about the status of a capture. Can be requested with a MAV_CMD_REQUEST_MESSAGE command.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
image_status	uint8_t		Current status of image capturing (0: idle, 1: capture in progress, 2: interval set but idle, 3: interval set and capture in progress)
video_status	uint8_t		Current status of video capturing (0: idle, 1: capture in progress)
image_interval	float	s	Image capture interval
recording_time_ms	uint32_t	ms	Elapsed time since recording started (0: Not supported/available). A GCS should compute recording time and use non-zero values of this field to correct any discrepancy.
available_capacity	float	MiB	Available storage capacity.
image_count**	int32_t		Total number of images captured ('forever', or until reset using MAV_CMD_STORAGE_FORMAT).

CAMERA_IMAGE_CAPTURED (#263

)

[Message] **(MAVLink 2)** Information about a captured image. This is emitted every time a message is captured.
MAV_CMD_REQUEST_MESSAGE can be used to (re)request this message for a specific sequence number or range of sequence numbers: MAV_CMD_REQUEST_MESSAGE.param2 indicates the sequence number the first image to send, or set to -1 to send the message for all sequence numbers. MAV_CMD_REQUEST_MESSAGE.param3 is used to specify a range of messages to send: set to 0 (default) to send just the the message for the sequence number in param 2, set to -1 to send the message for the sequence number in param 2 and all the following sequence numbers, set to the sequence number of the final message in the range.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
time_utc	uint64_t	us	Timestamp (time since UNIX epoch) in UTC. 0 for unknown.
camera_id	uint8_t		Deprecated/unused. Component IDs are used to differentiate multiple cameras.
lat	int32_t	degE7	Latitude where image was taken
lon	int32_t	degE7	Longitude where capture was taken
alt	int32_t	mm	Altitude (MSL) where image was taken
relative_alt	int32_t	mm	Altitude above ground
q	float[4]		Quaternion of camera orientation (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
image_index	int32_t		Zero based index of this image (i.e. a new image will have index CAMERA_CAPTURE_STATUS.image count -1)
capture_result	int8_t		Boolean indicating success (1) or failure (0) while capturing this image.
file_url	char[205]		URL of image taken. Either local storage or http://foo.jpg if camera provides an HTTP interface.

FLIGHT_INFORMATION (#264

)

[Message] **(MAVLink 2)** Information about flight since last arming. This can be requested using MAV_CMD_REQUEST_MESSAGE.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
arming_time_utc	uint64_t	us	Timestamp at arming (time since UNIX epoch) in UTC, 0 for unknown
takeoff_time_utc	uint64_t	us	Timestamp at takeoff (time since UNIX epoch) in UTC, 0 for unknown
flight_uuid	uint64_t		Universally unique identifier (UUID) of flight, should correspond to name of log files

MOUNT_ORIENTATION (#265

)

DEPRECATED: Replaced by MAV_CMD_DO_GIMBAL_MANAGER_PITCHYAW (2020-01).

This message is being superseded by MAV_CMD_DO_GIMBAL_MANAGER_PITCHYAW. The message can still be used to communicate with legacy gimbals implementing it.

[Message] **(MAVLink 2)** Orientation of a mount

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
roll	float	deg	Roll in global frame (set to NaN for invalid).
pitch	float	deg	Pitch in global frame (set to NaN for invalid).
yaw	float	deg	Yaw relative to vehicle (set to NaN for invalid).
yaw_absolute**	float	deg	Yaw in absolute frame relative to Earth's North, north is 0 (set to NaN for invalid).

LOGGING_DATA (#266

)

[Message] **(MAVLink 2)** A message containing logged data (see also MAV_CMD_LOGGING_START)

Field Name	Type	Units	Description
target_system	uint8_t		system ID of the target
target_component	uint8_t		component ID of the target
sequence	uint16_t		sequence number (can wrap)
length	uint8_t	bytes	data length
first_message_offset	uint8_t	bytes	offset into data where first message starts. This can be used for recovery, when a previous message got lost (set to UINT8_MAX if no start exists).
data	uint8_t[249]		logged data

LOGGING_DATA_ACKED (#267

)

[Message] **(MAVLink 2)** A message containing logged data which requires a LOGGING_ACK to be sent back

Field Name	Type	Units	Description
target_system	uint8_t		system ID of the target
target_component	uint8_t		component ID of the target
sequence	uint16_t		sequence number (can wrap)
length	uint8_t	bytes	data length
first_message_offset	uint8_t	bytes	offset into data where first message starts. This can be used for recovery, when a previous message got lost (set to UINT8_MAX if no start exists).
data	uint8_t[249]		logged data

LOGGING_ACK (#268

)

[Message] **(MAVLink 2)** An ack for a LOGGING_DATA_ACKED message

Field Name	Type	Description
target_system	uint8_t	system ID of the target
target_component	uint8_t	component ID of the target
sequence	uint16_t	sequence number (must match the one in LOGGING_DATA_ACKED)

VIDEO_STREAM_INFORMATION (#269

)

[Message] **(MAVLink 2)** Information about video stream. It may be requested using MAV_CMD_REQUEST_MESSAGE, where param2 indicates the video stream id: 0 for all streams, 1 for first, 2 for second, etc.

Field Name	Type	Units	Values	Description
stream_id	uint8_t			Video Stream ID (1 for first, 2 for second, etc.)
count	uint8_t			Number of streams available.
type	uint8_t		VIDEO_STREAM_TYPE	Type of stream.
flags	uint16_t		VIDEO_STREAM_STATUS_FLAGS	Bitmap of stream status flags.
framerate	float	Hz		Frame rate.
resolution_h	uint16_t	pix		Horizontal resolution.
resolution_v	uint16_t	pix		Vertical resolution.
bitrate	uint32_t	bits/s		Bit rate.
rotation	uint16_t	deg		Video image rotation clockwise.
hfov	uint16_t	deg		Horizontal Field of view.
name	char[32]			Stream name.
uri	char[160]			Video stream URI (TCP or RTSP URI ground station should connect to) or port number (UDP port ground station should listen to).

VIDEO_STREAM_STATUS (#270

)

[Message] **(MAVLink 2)** Information about the status of a video stream. It may be requested using MAV_CMD_REQUEST_MESSAGE.

Field Name	Type	Units	Values	Description
stream_id	uint8_t			Video Stream ID (1 for first, 2 for second, etc.)
flags	uint16_t		VIDEO_STREAM_STATUS_FLAGS	Bitmap of stream status flags
framerate	float	Hz		Frame rate
resolution_h	uint16_t	pix		Horizontal resolution
resolution_v	uint16_t	pix		Vertical resolution
bitrate	uint32_t	bits/s		Bit rate
rotation	uint16_t	deg		Video image rotation clockwise

hfov	uint16_t	deg	Horizontal Field of view

CAMERA_FOV_STATUS (#271

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Information about the field of view of a camera. Can be requested with a MAV_CMD_REQUEST_MESSAGE command.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
lat_camera	int32_t	degE7	Latitude of camera (INT32_MAX if unknown).
lon_camera	int32_t	degE7	Longitude of camera (INT32_MAX if unknown).
alt_camera	int32_t	mm	Altitude (MSL) of camera (INT32_MAX if unknown).
lat_image	int32_t	degE7	Latitude of center of image (INT32_MAX if unknown, INT32_MIN if at infinity, not intersecting with horizon).
lon_image	int32_t	degE7	Longitude of center of image (INT32_MAX if unknown, INT32_MIN if at infinity, not intersecting with horizon).
alt_image	int32_t	mm	Altitude (MSL) of center of image (INT32_MAX if unknown, INT32_MIN if at infinity, not intersecting with horizon).
q	float[4]		Quaternion of camera orientation (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
hfov	float	deg	Horizontal field of view (NaN if unknown).
vfov	float	deg	Vertical field of view (NaN if unknown).

CAMERA_TRACKING_IMAGE_STATUS (#275

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Camera tracking status, sent while in active tracking. Use MAV_CMD_SET_MESSAGE_INTERVAL to define message interval.

Field Name	Type	Values	Description
tracking_status	uint8_t	CAMERA_TRACKING_STATUS_FLAGS	Current tracking status
tracking_mode	uint8_t	CAMERA_TRACKING_MODE	Current tracking mode
target_data	uint8_t	CAMERA_TRACKING_TARGET_DATA	Defines location of target data
point_x	float		Current tracked point x value if CAMERA_TRACKING_MODE_POINT (normalized 0..1, 0 is left, 1 is right), NAN if unknown
point_y	float		Current tracked point y value if CAMERA_TRACKING_MODE_POINT (normalized 0..1, 0 is top, 1 is bottom), NAN if unknown
radius	float		Current tracked radius if CAMERA_TRACKING_MODE_POINT (normalized 0..1, 0 is image left, 1 is image right), NAN if unknown
rec_top_x	float		Current tracked rectangle top x value if CAMERA_TRACKING_MODE_RECTANGLE (normalized 0..1, 0 is left, 1 is right), NAN if unknown
rec_top_y	float		Current tracked rectangle top y value if CAMERA_TRACKING_MODE_RECTANGLE (normalized 0..1, 0 is top, 1 is bottom), NAN if unknown
rec_bottom_x	float		Current tracked rectangle bottom x value if CAMERA_TRACKING_MODE_RECTANGLE (normalized 0..1, 0 is left, 1 is right), NAN if unknown
rec_bottom_y	float		Current tracked rectangle bottom y value if CAMERA_TRACKING_MODE_RECTANGLE (normalized 0..1, 0 is top, 1 is bottom), NAN if unknown

CAMERA_TRACKING_GEO_STATUS (#276

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Camera tracking status, sent while in active tracking. Use MAV_CMD_SET_MESSAGE_INTERVAL to define message interval.

Field Name	Type	Units	Values	Description
tracking_status	uint8_t		CAMERA_TRACKING_STATUS_FLAGS	Current tracking status
lat	int32_t	degE7		Latitude of tracked object
lon	int32_t	degE7		Longitude of tracked object
alt	float	m		Altitude of tracked object(AMSL, WGS84)
h_acc	float	m		Horizontal accuracy. NAN if unknown
v_acc	float	m		Vertical accuracy. NAN if unknown
vel_n	float	m/s		North velocity of tracked object. NAN if unknown
vel_e	float	m/s		East velocity of tracked object. NAN if unknown
vel_d	float	m/s		Down velocity of tracked object. NAN if unknown
vel_acc	float	m/s		Velocity accuracy. NAN if unknown
dist	float	m		Distance between camera and tracked object. NAN if unknown
hdg	float	rad		Heading in radians, in NED. NAN if unknown
hdg_acc	float	rad		Accuracy of heading, in NED. NAN if unknown

GIMBAL_MANAGER_INFORMATION (#280

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Information about a high level gimbal manager. This message should be requested by a ground station using MAV_CMD_REQUEST_MESSAGE.

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
cap_flags	uint32_t		GIMBAL_MANAGER_CAP_FLAGS	Bitmap of gimbal capability flags.
gimbal_device_id	uint8_t			Gimbal device ID that this gimbal manager is responsible for.
roll_min	float	rad		Minimum hardware roll angle (positive: rolling to the right, negative: rolling to the left)
roll_max	float	rad		Maximum hardware roll angle (positive: rolling to the right, negative: rolling to the left)
pitch_min	float	rad		Minimum pitch angle (positive: up, negative: down)
pitch_max	float	rad		Maximum pitch angle (positive: up, negative: down)
yaw_min	float	rad		Minimum yaw angle (positive: to the right, negative: to the left)
yaw_max	float	rad		Maximum yaw angle (positive: to the right, negative: to the left)

GIMBAL_MANAGER_STATUS (#281

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Current status about a high level gimbal manager. This message should be broadcast at a low regular rate (e.g. 5Hz).

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
flags	uint32_t		GIMBAL_MANAGER_FLAGS	High level gimbal manager flags currently applied.
gimbal_device_id	uint8_t			Gimbal device ID that this gimbal manager is responsible for.
primary_control_sysid	uint8_t			System ID of MAVLink component with primary control, 0 for none.
primary_control_compid	uint8_t			Component ID of MAVLink component with primary control, 0 for none.
secondary_control_sysid	uint8_t			System ID of MAVLink component with secondary control, 0 for none.
secondary_control_compid	uint8_t			Component ID of MAVLink component with secondary control, 0 for none.

GIMBAL_MANAGER_SET_ATTITUDE (#282

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** High level message to control a gimbal's attitude. This message is to be sent to the gimbal manager (e.g. from a ground station). Angles and rates can be set to NaN according to use case.

Field Name	Type	Units	Values	Description
target_system	uint8_t			System ID
target_component	uint8_t			Component ID
flags	uint32_t		GIMBAL_MANAGER_FLAGS	High level gimbal manager flags to use.
gimbal_device_id	uint8_t			Component ID of gimbal device to address (or 1-6 for non-MAVLink gimbal), 0 for all gimbal device components. Send command multiple times for more than one gimbal (but not all gimbals).
q	float[4]			Quaternion components, w, x, y, z (1 0 0 0 is the null-rotation, the frame is depends on whether the flag GIMBAL_MANAGER_FLAGS_YAW_LOCK is set)
angular_velocity_x	float	rad/s		X component of angular velocity, positive is rolling to the right, NaN to be ignored.
angular_velocity_y	float	rad/s		Y component of angular velocity, positive is pitching up, NaN to be ignored.
angular_velocity_z	float	rad/s		Z component of angular velocity, positive is yawing to the right, NaN to be ignored.

GIMBAL_DEVICE_INFORMATION (#283

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Information about a low level gimbal. This message should be requested by the gimbal manager or a ground station using MAV_CMD_REQUEST_MESSAGE. The maximum angles and rates are the limits by hardware. However, the limits by software used are likely different/smaller and dependent on mode/settings/etc..

Field Name	Type	Units	Values	Description
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
vendor_name	char[32]			Name of the gimbal vendor.
model_name	char[32]			Name of the gimbal model.
custom_name	char[32]			Custom name of the gimbal given to it by the user.
firmware_version	uint32_t			Version of the gimbal firmware, encoded as: (Dev & 0xff) << 24
hardware_version	uint32_t			Version of the gimbal hardware, encoded as: (Dev & 0xff) << 24
uid	uint64_t			UID of gimbal hardware (0 if unknown).
cap_flags	uint16_t		GIMBAL_DEVICE_CAP_FLAGS	Bitmap of gimbal capability flags.
custom_cap_flags	uint16_t			Bitmap for use for gimbal-specific capability flags.
roll_min	float	rad		Minimum hardware roll angle (positive: rolling to the right, negative: rolling to the left)
roll_max	float	rad		Maximum hardware roll angle (positive: rolling to the right, negative:

			rolling to the left)
pitch_min	float	rad	Minimum hardware pitch angle (positive: up, negative: down)
pitch_max	float	rad	Maximum hardware pitch angle (positive: up, negative: down)
yaw_min	float	rad	Minimum hardware yaw angle (positive: to the right, negative: to the left)
yaw_max	float	rad	Maximum hardware yaw angle (positive: to the right, negative: to the left)

GIMBAL_DEVICE_SET_ATTITUDE (#284

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Low level message to control a gimbal device's attitude. This message is to be sent from the gimbal manager to the gimbal device component. Angles and rates can be set to NaN according to use case.

Field Name	Type	Units	Values	Description
target_system	uint8_t			System ID
target_component	uint8_t			Component ID
flags	uint16_t		GIMBAL_DEVICE_FLAGS	Low level gimbal flags.
q	float[4]			Quaternion components, w, x, y, z (1 0 0 0 is the null-rotation, the frame is depends on whether the flag GIMBAL_DEVICE_FLAGS_YAW_LOCK is set, set all fields to NaN if only angular velocity should be used)
angular_velocity_x	float	rad/s		X component of angular velocity, positive is rolling to the right, NaN to be ignored.
angular_velocity_y	float	rad/s		Y component of angular velocity, positive is pitching up, NaN to be ignored.
angular_velocity_z	float	rad/s		Z component of angular velocity, positive is yawing to the right, NaN to be ignored.

GIMBAL_DEVICE_ATTITUDE_STATUS (#285

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Message reporting the status of a gimbal device. This message should be broadcasted by a gimbal device component. The angles encoded in the quaternion are relative to absolute North if the flag GIMBAL_DEVICE_FLAGS_YAW_LOCK is set (roll: positive is rolling to the right, pitch: positive is pitching up, yaw is turn to the right) or relative to the vehicle heading if the flag is not set. This message should be broadcast at a low regular rate (e.g. 10Hz).

Field Name	Type	Units	Values	Description
target_system	uint8_t			System ID
target_component	uint8_t			Component ID
time_boot_ms	uint32_t	ms		Timestamp (time since system boot).
flags	uint16_t		GIMBAL_DEVICE_FLAGS	Current gimbal flags set.
q	float[4]			Quaternion components, w, x, y, z (1 0 0 0 is the null-rotation, the frame is depends on whether the flag GIMBAL_DEVICE_FLAGS_YAW_LOCK is set)
angular_velocity_x	float	rad/s		X component of angular velocity (NaN if unknown)
angular_velocity_y	float	rad/s		Y component of angular velocity (NaN if unknown)
angular_velocity_z	float	rad/s		Z component of angular velocity (NaN if unknown)
failure_flags	uint32_t		GIMBAL_DEVICE_ERROR_FLAGS	Failure flags (0 for no failure)

AUTOPILOT_STATE_FOR_GIMBAL_DEVICE (#286

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Low level message containing autopilot state relevant for a gimbal device. This message is to be sent from the gimbal manager to the gimbal device component. The data of this message server for the gimbal's estimator corrections in particular horizon compensation, as well as the autopilot's control intention e.g. feed forward angular control in z-axis.

Field Name	Type	Units	Values	Description
target_system	uint8_t			System ID
target_component	uint8_t			Component ID
time_boot_us	uint64_t	us		Timestamp (time since system boot).
q	float[4]			Quaternion components of autopilot attitude: w, x, y, z (1 0 0 0 is the null-rotation, Hamilton convention).
q_estimated_delay_us	uint32_t	us		Estimated delay of the attitude data.
vx	float	m/s		X Speed in NED (North, East, Down).
vy	float	m/s		Y Speed in NED (North, East, Down).
vz	float	m/s		Z Speed in NED (North, East, Down).
v_estimated_delay_us	uint32_t	us		Estimated delay of the speed data.
feed_forward_angular_velocity_z	float	rad/s		Feed forward Z component of angular velocity, positive is yawing to the right, NaN to be ignored. This is to indicate if the autopilot is actively yawing.
estimator_status	uint16_t		ESTIMATOR_STATUS_FLAGS	Bitmap indicating which estimator outputs are valid.
landed_state	uint8_t		MAV_LANDED_STATE	The landed state. Is set to MAV_LANDED_STATE_UNDEFINED if landed state is unknown.

GIMBAL_MANAGER_SET_PITCHYAW (#287

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** High level message to control a gimbal's pitch and yaw angles. This message is to be sent to the gimbal manager (e.g. from a ground station). Angles and rates can be set to NaN according to use case.

Field Name	Type	Units	Values	Description
target_system	uint8_t			System ID
target_component	uint8_t			Component ID
flags	uint32_t		GIMBAL_MANAGER_FLAGS	High level gimbal manager flags to use.
gimbal_device_id	uint8_t			Component ID of gimbal device to address (or 1-6 for non-MAVLink gimbal), 0 for all gimbal device components. Send command multiple times for more than one gimbal (but not all gimbals).
pitch	float	rad		Pitch angle (positive: up, negative: down, NaN to be ignored).
yaw	float	rad		Yaw angle (positive: to the right, negative: to the left, NaN to be ignored).
pitch_rate	float	rad/s		Pitch angular rate (positive: up, negative: down, NaN to be ignored).
yaw_rate	float	rad/s		Yaw angular rate (positive: to the right, negative: to the left, NaN to be ignored).

GIMBAL_MANAGER_SET_MANUAL_CONTROL (#288

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** High level message to control a gimbal manually. The angles or angular rates are unitless; the actual rates will depend on internal gimbal manager settings/configuration (e.g. set by parameters). This message is to be sent to the gimbal manager (e.g. from a ground station). Angles and rates can be set to NaN according to use case.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
flags	uint32_t	GIMBAL_MANAGER_FLAGS	High level gimbal manager flags.
gimbal_device_id	uint8_t		Component ID of gimbal device to address (or 1-6 for non-MAVLink gimbal), 0 for all gimbal device components. Send command multiple times for more than one gimbal (but not all gimbals).
pitch	float		Pitch angle unitless (-1..1, positive: up, negative: down, NaN to be ignored).
yaw	float		Yaw angle unitless (-1..1, positive: to the right, negative: to the left, NaN to be ignored).
pitch_rate	float		Pitch angular rate unitless (-1..1, positive: up, negative: down, NaN to be ignored).
yaw_rate	float		Yaw angular rate unitless (-1..1, positive: to the right, negative: to the left, NaN to be ignored).

ESC_INFO (#290

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** ESC information for lower rate streaming. Recommended streaming rate 1Hz. See ESC_STATUS for higher-rate ESC data.

Field Name	Type	Units	Values	Description
index	uint8_t			Index of the first ESC in this message. minValue = 0, maxValue = 60, increment = 4.
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude the number.
counter	uint16_t			Counter of data packets received.
count	uint8_t			Total number of ESCs in all messages of this type. Message fields with an index higher than this should be ignored because they contain invalid data.
connection_type	uint8_t		ESC_CONNECTION_TYPE	Connection type protocol for all ESC.
info	uint8_t			Information regarding online/offline status of each ESC.
failure_flags	uint16_t[4]		ESC_FAILURE_FLAGS	Bitmap of ESC failure flags.
error_count	uint32_t[4]			Number of reported errors by each ESC since boot.
temperature	int16_t[4]	cdegC		Temperature of each ESC. INT16_MAX: if data not supplied by ESC.

ESC_STATUS (#291

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** ESC information for higher rate streaming. Recommended streaming rate is ~10 Hz. Information that changes more slowly is sent in ESC_INFO. It should typically only be streamed on high-bandwidth links (i.e. to a companion computer).

Field Name	Type	Units	Description
index	uint8_t		Index of the first ESC in this message. minValue = 0, maxValue = 60, increment = 4.
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude the number.
rpm	int32_t[4]	rpm	Reported motor RPM from each ESC (negative for reverse rotation).
voltage	float[4]	V	Voltage measured from each ESC.
current	float[4]	A	Current measured from each ESC.

WIFI_CONFIG_AP (#299

)

[Message] **(MAVLink 2)** Configure WiFi AP SSID, password, and mode. This message is re-emitted as an acknowledgement by the AP. The message may also be explicitly requested using MAV_CMD_REQUEST_MESSAGE

Field Name	Type	Values	Description
ssid	char[32]		Name of Wi-Fi network (SSID). Blank to leave it unchanged when setting. Current SSID when sent back as a response.
password	char[64]		Password. Blank for an open AP. MD5 hash when message is sent back as a response.
mode**	int8_t	WIFI_CONFIG_AP_MODE	WiFi Mode.
response**	int8_t	WIFI_CONFIG_AP_RESPONSE	Message acceptance response (sent back to GS).

AIS_VESSEL (#301

)

[Message] **(MAVLink 2)** The location and information of an AIS vessel

Field Name	Type	Units	Values	Description
MMSI	uint32_t			Mobile Marine Service Identifier, 9 decimal digits
lat	int32_t	degE7		Latitude
lon	int32_t	degE7		Longitude
COG	uint16_t	cdeg		Course over ground
heading	uint16_t	cdeg		True heading
velocity	uint16_t	cm/s		Speed over ground
turn_rate	int8_t	cdeg/s		Turn rate
navigational_status	uint8_t		AIS_NAV_STATUS	Navigational status
type	uint8_t		AIS_TYPE	Type of vessels
dimension_bow	uint16_t	m		Distance from lat/lon location to bow
dimension_stern	uint16_t	m		Distance from lat/lon location to stern
dimension_port	uint8_t	m		Distance from lat/lon location to port side
dimension_starboard	uint8_t	m		Distance from lat/lon location to starboard side
callsign	char[7]			The vessel callsign
name	char[20]			The vessel name
tslc	uint16_t	s		Time since last communication in seconds
flags	uint16_t		AIS_FLAGS	Bitmask to indicate various statuses including valid data fields

UAVCAN_NODE_STATUS (#310

)

[Message] **(MAVLink 2)** General status information of an UAVCAN node. Please refer to the definition of the UAVCAN message "uavcan.protocol.NodeStatus" for the background information. The UAVCAN specification is available at http://uavcan.org.

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
uptime_sec	uint32_t	s		Time since the start-up of the node.
health	uint8_t		UAVCAN_NODE_HEALTH	Generalized node health status.
mode	uint8_t		UAVCAN_NODE_MODE	Generalized operating mode.
sub_mode	uint8_t			Not used currently.
vendor_specific_status_code	uint16_t			Vendor-specific status information.

UAVCAN_NODE_INFO (#311

)

[Message] **(MAVLink 2)** General information describing a particular UAVCAN node. Please refer to the definition of the UAVCAN service "uavcan.protocol.GetNodeInfo" for the background information. This message should be emitted by the system whenever a new node appears online, or an existing node reboots. Additionally, it can be emitted upon request from the other end of the MAVLink channel (see MAV_CMD_UAVCAN_GET_NODE_INFO). It is also not prohibited to emit this message unconditionally at a low frequency. The UAVCAN specification is available at http://uavcan.org.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
uptime_sec	uint32_t	s	Time since the start-up of the node.
name	char[80]		Node name string. For example, "sapog.px4.io".
hw_version_major	uint8_t		Hardware major version number.
hw_version_minor	uint8_t		Hardware minor version number.
hw_unique_id	uint8_t[16]		Hardware unique 128-bit ID.
sw_version_major	uint8_t		Software major version number.
sw_version_minor	uint8_t		Software minor version number.
sw_vcs_commit	uint32_t		Version control system (VCS) revision identifier (e.g. git short commit hash). 0 if unknown.

PARAM_EXT_REQUEST_READ (#320

)

[Message] **(MAVLink 2)** Request to read the value of a parameter with either the param_id string id or param_index. PARAM_EXT_VALUE should be emitted in response.

Field Name	Type	Description
target_system	uint8_t	System ID

target_component	uint8_t	Component ID
param_id	char[16]	Parameter id, terminated by NULL if the length is less than 16 human-readable chars and WITHOUT null termination (NULL) byte if the length is exactly 16 chars - applications have to provide 16+1 bytes storage if the ID is stored as string
param_index	int16_t	Parameter index. Set to -1 to use the Parameter ID field as identifier (else param_id will be ignored)

PARAM_EXT_REQUEST_LIST (#321

)

[Message] **(MAVLink 2)** Request all parameters of this component. All parameters should be emitted in response as PARAM_EXT_VALUE.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID

PARAM_EXT_VALUE (#322

)

[Message] **(MAVLink 2)** Emit the value of a parameter. The inclusion of param_count and param_index in the message allows the recipient to keep track of received parameters and allows them to re-request missing parameters after a loss or timeout.

Field Name	Type	Values	Description
param_id	char[16]		Parameter id, terminated by NULL if the length is less than 16 human-readable chars and WITHOUT null termination (NULL) byte if the length is exactly 16 chars - applications have to provide 16+1 bytes storage if the ID is stored as string
param_value	char[128]		Parameter value
param_type	uint8_t	MAV_PARAM_EXT_TYPE	Parameter type.
param_count	uint16_t		Total number of parameters
param_index	uint16_t		Index of this parameter

PARAM_EXT_SET (#323

)

[Message] **(MAVLink 2)** Set a parameter value. In order to deal with message loss (and retransmission of PARAM_EXT_SET), when setting a parameter value and the new value is the same as the current value, you will immediately get a PARAM_ACK_ACCEPTED response. If the current state is PARAM_ACK_IN_PROGRESS, you will accordingly receive a PARAM_ACK_IN_PROGRESS in response.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
param_id	char[16]		Parameter id, terminated by NULL if the length is less than 16 human-readable chars and WITHOUT null termination (NULL) byte if the length is exactly 16 chars - applications have to provide 16+1 bytes storage if the ID is stored as string
param_value	char[128]		Parameter value
param_type	uint8_t	MAV_PARAM_EXT_TYPE	Parameter type.

PARAM_EXT_ACK (#324

)

[Message] **(MAVLink 2)** Response from a PARAM_EXT_SET message.

Field Name	Type	Values	Description
param_id	char[16]		Parameter id, terminated by NULL if the length is less than 16 human-readable chars and WITHOUT null termination (NULL) byte if the length is exactly 16 chars - applications have to provide 16+1 bytes storage if the ID is stored as string
param_value	char[128]		Parameter value (new value if PARAM_ACK_ACCEPTED, current value otherwise)
param_type	uint8_t	MAV_PARAM_EXT_TYPE	Parameter type.
param_result	uint8_t	PARAM_ACK	Result code.

OBSTACLE_DISTANCE (#330

)

[Message] **(MAVLink 2)** Obstacle distances in front of the sensor, starting from the left in increment degrees to the right

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
sensor_type	uint8_t		MAV_DISTANCE_SENSOR	Class id of the distance sensor type.
distances	uint16_t[72]	cm		Distance of obstacles around the vehicle with index 0 corresponding to north + angle_offset, unless otherwise specified in the frame. A value of 0 is valid and means that the obstacle is practically touching the sensor. A value of max_distance +1 means no obstacle is present. A value of UINT16_MAX for unknown/not used. In a array element, one unit corresponds to 1cm.
increment	uint8_t	deg		Angular width in degrees of each array element. Increment direction is clockwise. This field is ignored if increment_f is non-zero.
min_distance	uint16_t	cm		Minimum distance the sensor can measure.
max_distance	uint16_t	cm		Maximum distance the sensor can measure.
increment_f**	float	deg		Angular width in degrees of each array element as a float. If non-zero then this value is used instead of the uint8_t increment field. Positive is clockwise direction, negative is counter-clockwise.
				Relative angle offset of the 0-index element in the distances array. Value of

angle_offset**	float	deg		0 corresponds to forward. Positive is clockwise direction, negative is counter-clockwise.
frame**	uint8_t		MAV_FRAME	Coordinate frame of reference for the yaw rotation and offset of the sensor data. Defaults to MAV_FRAME_GLOBAL, which is north aligned. For body-mounted sensors use MAV_FRAME_BODY_FRD, which is vehicle front aligned.

ODOMETRY (#331

)

[Message] **(MAVLink 2)** Odometry message to communicate odometry information with an external interface. Fits ROS REP 147 standard for aerial vehicles (http://www.ros.org/reps/rep-0147.html).

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
frame_id	uint8_t		MAV_FRAME	Coordinate frame of reference for the pose data.
child_frame_id	uint8_t		MAV_FRAME	Coordinate frame of reference for the velocity in free space (twist) data.
x	float	m		X Position
y	float	m		Y Position
z	float	m		Z Position
q	float[4]			Quaternion components, w, x, y, z (1 0 0 0 is the null-rotation)
vx	float	m/s		X linear speed
vy	float	m/s		Y linear speed
vz	float	m/s		Z linear speed
rollspeed	float	rad/s		Roll angular speed
pitchspeed	float	rad/s		Pitch angular speed
yawspeed	float	rad/s		Yaw angular speed
pose_covariance	float[21]			Row-major representation of a 6x6 pose cross-covariance matrix upper right triangle (states: x, y, z, roll, pitch, yaw; first six entries are the first ROW, next five entries are the second ROW, etc.). If unknown, assign NaN value to first element in the array.
velocity_covariance	float[21]			Row-major representation of a 6x6 velocity cross-covariance matrix upper right triangle (states: vx, vy, vz, rollspeed, pitchspeed, yawspeed; first six entries are the first ROW, next five entries are the second ROW, etc.). If unknown, assign NaN value to first element in the array.
reset_counter**	uint8_t			Estimate reset counter. This should be incremented when the estimate resets in any of the dimensions (position, velocity, attitude, angular speed). This is designed to be used when e.g an external SLAM system detects a loop-closure and the estimate jumps.
estimator_type**	uint8_t		MAV_ESTIMATOR_TYPE	Type of estimator that is providing the odometry.
quality**	int8_t	%		Optional odometry quality metric as a percentage. -1 = odometry has failed, 0 = unknown/unset quality, 1 = worst quality, 100 = best quality

TRAJECTORY_REPRESENTATION_WAYPOINTS (#332

)

[Message] **(MAVLink 2)** Describe a trajectory using an array of up-to 5 waypoints in the local frame (MAV_FRAME_LOCAL_NED).

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
valid_points	uint8_t			Number of valid points (up-to 5 waypoints are possible)
pos_x	float[5]	m		X-coordinate of waypoint, set to NaN if not being used
pos_y	float[5]	m		Y-coordinate of waypoint, set to NaN if not being used
pos_z	float[5]	m		Z-coordinate of waypoint, set to NaN if not being used
vel_x	float[5]	m/s		X-velocity of waypoint, set to NaN if not being used
vel_y	float[5]	m/s		Y-velocity of waypoint, set to NaN if not being used
vel_z	float[5]	m/s		Z-velocity of waypoint, set to NaN if not being used
acc_x	float[5]	m/s/s		X-acceleration of waypoint, set to NaN if not being used
acc_y	float[5]	m/s/s		Y-acceleration of waypoint, set to NaN if not being used
acc_z	float[5]	m/s/s		Z-acceleration of waypoint, set to NaN if not being used
pos_yaw	float[5]	rad		Yaw angle, set to NaN if not being used
vel_yaw	float[5]	rad/s		Yaw rate, set to NaN if not being used
command	uint16_t[5]		MAV_CMD	MAV_CMD command id of waypoint, set to UINT16_MAX if not being used.

TRAJECTORY_REPRESENTATION_BEZIER (#333

)

[Message] **(MAVLink 2)** Describe a trajectory using an array of up-to 5 bezier control points in the local frame (MAV_FRAME_LOCAL_NED).

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
valid_points	uint8_t		Number of valid control points (up-to 5 points are possible)
pos_x	float[5]	m	X-coordinate of bezier control points. Set to NaN if not being used
pos_y	float[5]	m	Y-coordinate of bezier control points. Set to NaN if not being used
pos_z	float[5]	m	Z-coordinate of bezier control points. Set to NaN if not being used
delta	float[5]	s	Bezier time horizon. Set to NaN if velocity/acceleration should not be incorporated
pos_yaw	float[5]	rad	Yaw. Set to NaN for unchanged

CELLULAR_STATUS (#334

)

[Message] **(MAVLink 2)** Report current used cellular network status

Field Name	Type	Values	Description
status	uint8_t	CELLULAR_STATUS_FLAG	Cellular modem status
failure_reason	uint8_t	CELLULAR_NETWORK_FAILED_REASON	Failure reason when status in in CELLUAR_STATUS_FAILED
type	uint8_t	CELLULAR_NETWORK_RADIO_TYPE	Cellular network radio type: gsm, cdma, lte...
quality	uint8_t		Signal quality in percent. If unknown, set to UINT8_MAX
mcc	uint16_t		Mobile country code. If unknown, set to UINT16_MAX
mnc	uint16_t		Mobile network code. If unknown, set to UINT16_MAX
lac	uint16_t		Location area code. If unknown, set to 0

ISBD_LINK_STATUS (#335

)

[Message] **(MAVLink 2)** Status of the Iridium SBD link.

Field Name	Type	Units	Description
timestamp	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
last_heartbeat	uint64_t	us	Timestamp of the last successful sbd session. The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
failed_sessions	uint16_t		Number of failed SBD sessions.
successful_sessions	uint16_t		Number of successful SBD sessions.
signal_quality	uint8_t		Signal quality equal to the number of bars displayed on the ISU signal strength indicator. Range is 0 to 5, where 0 indicates no signal and 5 indicates maximum signal strength.
ring_pending	uint8_t		1: Ring call pending, 0: No call pending.
tx_session_pending	uint8_t		1: Transmission session pending, 0: No transmission session pending.
rx_session_pending	uint8_t		1: Receiving session pending, 0: No receiving session pending.

CELLULAR_CONFIG (#336

)

[Message] **(MAVLink 2)** Configure cellular modems. This message is re-emitted as an acknowledgement by the modem. The message may also be explicitly requested using MAV_CMD_REQUEST_MESSAGE.

Field Name	Type	Values	Description
enable_lte	uint8_t		Enable/disable LTE. 0: setting unchanged, 1: disabled, 2: enabled. Current setting when sent back as a response.
enable_pin	uint8_t		Enable/disable PIN on the SIM card. 0: setting unchanged, 1: disabled, 2: enabled. Current setting when sent back as a response.
pin	char[16]		PIN sent to the SIM card. Blank when PIN is disabled. Empty when message is sent back as a response.
new_pin	char[16]		New PIN when changing the PIN. Blank to leave it unchanged. Empty when message is sent back as a response.
apn	char[32]		Name of the cellular APN. Blank to leave it unchanged. Current APN when sent back as a response.
puk	char[16]		Required PUK code in case the user failed to authenticate 3 times with the PIN. Empty when message is sent back as a response.
roaming	uint8_t		Enable/disable roaming. 0: setting unchanged, 1: disabled, 2: enabled. Current setting when sent back as a response.
response	uint8_t	CELLULAR_CONFIG_RESPONSE	Message acceptance response (sent back to GS).

RAW_RPM (#339

)

[Message] **(MAVLink 2)** RPM sensor data message.

Field Name	Type	Units	Description
index	uint8_t		Index of this RPM sensor (0-indexed)
frequency	float	rpm	Indicated rate

UTM_GLOBAL_POSITION (#340

)

[Message] **(MAVLink 2)** The global position resulting from GPS and sensor fusion.

Field Name	Type	Units	Values	Description
time	uint64_t	us		Time of applicability of position (microseconds since UNIX epoch).
uas_id	uint8_t[18]			Unique UAS ID.
lat	int32_t	degE7		Latitude (WGS84)
lon	int32_t	degE7		Longitude (WGS84)
alt	int32_t	mm		Altitude (WGS84)
relative_alt	int32_t	mm		Altitude above ground
vx	int16_t	cm/s		Ground X speed (latitude, positive north)
vy	int16_t	cm/s		Ground Y speed (longitude, positive east)
vz	int16_t	cm/s		Ground Z speed (altitude, positive down)
h_acc	uint16_t	mm		Horizontal position uncertainty (standard deviation)
v_acc	uint16_t	mm		Altitude uncertainty (standard deviation)
vel_acc	uint16_t	cm/s		Speed uncertainty (standard deviation)
next_lat	int32_t	degE7		Next waypoint, latitude (WGS84)
next_lon	int32_t	degE7		Next waypoint, longitude (WGS84)
next_alt	int32_t	mm		Next waypoint, altitude (WGS84)
update_rate	uint16_t	cs		Time until next update. Set to 0 if unknown or in data driven mode.
flight_state	uint8_t		UTM_FLIGHT_STATE	Flight state
flags	uint8_t		UTM_DATA_AVAIL_FLAGS	Bitwise OR combination of the data available flags.

DEBUG_FLOAT_ARRAY (#350

)

[Message] **(MAVLink 2)** Large debug/prototyping array. The message uses the maximum available payload for data. The array_id and name fields are used to discriminate between messages in code and in user interfaces (respectively). Do not use in production code.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
name	char[10]		Name, for human-friendly display in a Ground Control Station
array_id	uint16_t		Unique ID used to discriminate between arrays
data**	float[58]		data

ORBIT_EXECUTION_STATUS (#360

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Vehicle status report that is sent out while orbit execution is in progress (see MAV_CMD_DO_ORBIT).

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
radius	float	m		Radius of the orbit circle. Positive values orbit clockwise, negative values orbit counter-clockwise.
frame	uint8_t		MAV_FRAME	The coordinate system of the fields: x, y, z.
x	int32_t			X coordinate of center point. Coordinate system depends on frame field: local = x position in meters * 1e4, global = latitude in degrees * 1e7.
y	int32_t			Y coordinate of center point. Coordinate system depends on frame field: local = x position in meters * 1e4, global = latitude in degrees * 1e7.
z	float	m		Altitude of center point. Coordinate system depends on frame field.

SMART_BATTERY_INFO (#370

)

[Message] **(MAVLink 2)** Smart Battery information (static/infrequent update). Use for updates from: smart battery to flight stack, flight stack to GCS. Use BATTERY_STATUS for smart battery frequent updates.

Field Name	Type	Units	Values	Description
id	uint8_t			Battery ID
battery_function	uint8_t		MAV_BATTERY_FUNCTION	Function of the battery
type	uint8_t		MAV_BATTERY_TYPE	Type (chemistry) of the battery
capacity_full_specification	int32_t	mAh		Capacity when full according to manufacturer, -1: field not provided.
capacity_full	int32_t	mAh		Capacity when full (accounting for battery degradation), -1: field not provided.
cycle_count	uint16_t			Charge/discharge cycle count. UINT16_MAX: field not provided.
serial_number	char[16]			Serial number in ASCII characters, 0 terminated. All 0: field not provided.
device_name	char[50]			Static device name in ASCII characters, 0 terminated. All 0: field not provided. Encode as manufacturer name then product name separated using an underscore.
weight	uint16_t	g		Battery weight. 0: field not provided.

discharge_minimum_voltage	uint16_t	mV	Minimum per-cell voltage when discharging. If not supplied set to UINT16_MAX value.
charging_minimum_voltage	uint16_t	mV	Minimum per-cell voltage when charging. If not supplied set to UINT16_MAX value.
resting_minimum_voltage	uint16_t	mV	Minimum per-cell voltage when resting. If not supplied set to UINT16_MAX value.
charging_maximum_voltage**	uint16_t	mV	Maximum per-cell voltage when charged. 0: field not provided.
cells_in_series**	uint8_t		Number of battery cells in series. 0: field not provided.
discharge_maximum_current**	uint32_t	mA	Maximum pack discharge current. 0: field not provided.
discharge_maximum_burst_current**	uint32_t	mA	Maximum pack discharge burst current. 0: field not provided.
manufacture_date**	char[11]		Manufacture date (DD/MM/YYYY) in ASCII characters, 0 terminated. All 0: field not provided.

GENERATOR_STATUS (#373

)

[Message] **(MAVLink 2)** Telemetry of power generation system. Alternator or mechanical generator.

Field Name	Type	Units	Values	Description
status	uint64_t		MAV_GENERATOR_STATUS_FLAG	Status flags.
generator_speed	uint16_t	rpm		Speed of electrical generator or alternator. UINT16_MAX: field not provided.
battery_current	float	A		Current into/out of battery. Positive for out. Negative for in. NaN: field not provided.
load_current	float	A		Current going to the UAV. If battery current not available this is the DC current from the generator. Positive for out. Negative for in. NaN: field not provided
power_generated	float	W		The power being generated. NaN: field not provided
bus_voltage	float	V		Voltage of the bus seen at the generator, or battery bus if battery bus is controlled by generator and at a different voltage to main bus.
rectifier_temperature	int16_t	degC		The temperature of the rectifier or power converter. INT16_MAX: field not provided.
bat_current_setpoint	float	A		The target battery current. Positive for out. Negative for in. NaN: field not provided
generator_temperature	int16_t	degC		The temperature of the mechanical motor, fuel cell core or generator. INT16_MAX: field not provided.
runtime	uint32_t	s		Seconds this generator has run since it was rebooted. UINT32_MAX: field not provided.
time_until_maintenance	int32_t	s		Seconds until this generator requires maintenance. A negative value indicates maintenance is past-due. INT32_MAX: field not provided.

ACTUATOR_OUTPUT_STATUS (#375

)

[Message] **(MAVLink 2)** The raw values of the actuator outputs (e.g. on Pixhawk, from MAIN, AUX ports). This message supersedes SERVO_OUTPUT_RAW.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (since system boot).
active	uint32_t		Active outputs
actuator	float[32]		Servo / motor output array values. Zero values indicate unused channels.

TIME_ESTIMATE_TO_TARGET (#380

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Time/duration estimates for various events and actions given the current vehicle state and position.

Field Name	Type	Units	Description
safe_return	int32_t	s	Estimated time to complete the vehicle's configured "safe return" action from its current position (e.g. RTL, Smart RTL, etc.). -1 indicates that the vehicle is landed, or that no time estimate available.
land	int32_t	s	Estimated time for vehicle to complete the LAND action from its current position. -1 indicates that the vehicle is landed, or that no time estimate available.
mission_next_item	int32_t	s	Estimated time for reaching/completing the currently active mission item. -1 means no time estimate available.
mission_end	int32_t	s	Estimated time for completing the current mission. -1 means no mission active and/or no estimate available.
commanded_action	int32_t	s	Estimated time for completing the current commanded action (i.e. Go To, Takeoff, Land, etc.). -1 means no action active and/or no estimate available.

TUNNEL (#385

)

[Message] **(MAVLink 2)** Message for transporting "arbitrary" variable-length data from one component to another (broadcast is not forbidden, but discouraged). The encoding of the data is usually extension specific, i.e. determined by the source, and is usually not documented as part of the MAVLink specification.

Field Name	Type	Values	Description
target_system	uint8_t		System ID (can be 0 for broadcast, but this is discouraged)

target_component	uint8_t		Component ID (can be 0 for broadcast, but this is discouraged)
payload_type	uint16_t	MAV_TUNNEL_PAYLOAD_TYPE	A code that identifies the content of the payload (0 for unknown, which is the default). If this code is less than 32768, it is a 'registered' payload type and the corresponding code should be added to the MAV_TUNNEL_PAYLOAD_TYPE enum. Software creators can register blocks of types as needed. Codes greater than 32767 are considered local experiments and should not be checked in to any widely distributed codebase.
payload_length	uint8_t		Length of the data transported in payload
payload	uint8_t[128]		Variable length payload. The payload length is defined by payload_length. The entire content of this block is opaque unless you understand the encoding specified by payload_type.

CAN_FRAME (#386

)

[Message] **(MAVLink 2)** A forwarded CAN frame as requested by MAV_CMD_CAN_FORWARD.

Field Name	Type	Description
target_system	uint8_t	System ID.
target_component	uint8_t	Component ID.
bus	uint8_t	Bus number
len	uint8_t	Frame length
id	uint32_t	Frame ID
data	uint8_t[8]	Frame data

ONBOARD_COMPUTER_STATUS (#390

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Hardware status sent by an onboard computer.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (UNIX Epoch time or time since system boot). The receiving end can infer timestamp format (since 1.1.1970 or since system boot) by checking for the magnitude of the number.
uptime	uint32_t	ms	Time since system boot.
type	uint8_t		Type of the onboard computer: 0: Mission computer primary, 1: Mission computer backup 1, 2: Mission computer backup 2, 3: Compute node, 4-5: Compute spares, 6-9: Payload computers.
cpu_cores	uint8_t[8]		CPU usage on the component in percent (100 - idle). A value of UINT8_MAX implies the field is unused.
cpu_combined	uint8_t[10]		Combined CPU usage as the last 10 slices of 100 MS (a histogram). This allows to identify spikes in load that max out the system, but only for a short amount of time. A value of UINT8_MAX implies the field is unused.
gpu_cores	uint8_t[4]		GPU usage on the component in percent (100 - idle). A value of UINT8_MAX implies the field is unused.
gpu_combined	uint8_t[10]		Combined GPU usage as the last 10 slices of 100 MS (a histogram). This allows to identify spikes in load that max out the system, but only for a short amount of time. A value of UINT8_MAX implies the field is unused.
temperature_board	int8_t	degC	Temperature of the board. A value of INT8_MAX implies the field is unused.
temperature_core	int8_t[8]	degC	Temperature of the CPU core. A value of INT8_MAX implies the field is unused.
fan_speed	int16_t[4]	rpm	Fan speeds. A value of INT16_MAX implies the field is unused.
ram_usage	uint32_t	MiB	Amount of used RAM on the component system. A value of UINT32_MAX implies the field is unused.
ram_total	uint32_t	MiB	Total amount of RAM on the component system. A value of UINT32_MAX implies the field is unused.
storage_type	uint32_t[4]		Storage type: 0: HDD, 1: SSD, 2: EMMC, 3: SD card (non-removable), 4: SD card (removable). A value of UINT32_MAX implies the field is unused.
storage_usage	uint32_t[4]	MiB	Amount of used storage space on the component system. A value of UINT32_MAX implies the field is unused.
storage_total	uint32_t[4]	MiB	Total amount of storage space on the component system. A value of UINT32_MAX implies the field is unused.
link_type	uint32_t[6]		Link type: 0-9: UART, 10-19: Wired network, 20-29: Wifi, 30-39: Point-to-point proprietary, 40-49: Mesh proprietary
link_tx_rate	uint32_t[6]	KiB/s	Network traffic from the component system. A value of UINT32_MAX implies the field is unused.
link_rx_rate	uint32_t[6]	KiB/s	Network traffic to the component system. A value of UINT32_MAX implies the field is unused.
link_tx_max	uint32_t[6]	KiB/s	Network capacity from the component system. A value of UINT32_MAX implies the field is unused.
link_rx_max	uint32_t[6]	KiB/s	Network capacity to the component system. A value of UINT32_MAX implies the field is unused.

COMPONENT_INFORMATION (#395

)

DEPRECATED: Replaced by COMPONENT_METADATA (2022-04).

[Message] **(MAVLink 2)** Component information message, which may be requested using MAV_CMD_REQUEST_MESSAGE.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
general_metadata_file_crc	uint32_t		CRC32 of the general metadata file (general_metadata_uri).
general_metadata_uri	char[100]		MAVLink FTP URI for the general metadata file (COMP_METADATA_TYPE_GENERAL), which may be compressed with xz. The file contains general component metadata, and may contain URI links for

		additional metadata (see COMP_METADATA_TYPE). The information is static from boot, and may be generated at compile time. The string needs to be zero terminated.
peripherals_metadata_file_crc	uint32_t	CRC32 of peripherals metadata file (peripherals_metadata_uri).
peripherals_metadata_uri	char[100]	(Optional) MAVLink FTP URI for the peripherals metadata file (COMP_METADATA_TYPE_PERIPHERALS), which may be compressed with xz. This contains data about "attached components" such as UAVCAN nodes. The peripherals are in a separate file because the information must be generated dynamically at runtime. The string needs to be zero terminated.

|

COMPONENT_METADATA (#397

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Component metadata message, which may be requested using MAV_CMD_REQUEST_MESSAGE.

This contains the MAVLink FTP URI and CRC for the component's general metadata file. The file must be hosted on the component, and may be xz compressed. The file CRC can be used for file caching.

The general metadata file can be read to get the locations of other metadata files (COMP_METADATA_TYPE) and translations, which may be hosted either on the vehicle or the internet. For more information see: https://mavlink.io/en/services/component_information.html.

Note: Camera components should use CAMERA_INFORMATION instead, and autopilots may use both this message and AUTOPILOT_VERSION.

Field Name	Type	Units	Description
time_boot_ms	uint32_t	ms	Timestamp (time since system boot).
file_crc	uint32_t		CRC32 of the general metadata file.
uri	char[100]		MAVLink FTP URI for the general metadata file (COMP_METADATA_TYPE_GENERAL), which may be compressed with xz. The file contains general component metadata, and may contain URI links for additional metadata (see COMP_METADATA_TYPE). The information is static from boot, and may be generated at compile time. The string needs to be zero terminated.

PLAY_TUNE_V2 (#400

)

[Message] **(MAVLink 2)** Play vehicle tone/tune (buzzer). Supersedes message PLAY_TUNE.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
format	uint32_t	TUNE_FORMAT	Tune format
tune	char[248]		Tune definition as a NULL-terminated string.

SUPPORTED_TUNES (#401

)

[Message] **(MAVLink 2)** Tune formats supported by vehicle. This should be emitted as response to MAV_CMD_REQUEST_MESSAGE.

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
format	uint32_t	TUNE_FORMAT	Bitfield of supported tune formats.

EVENT (#410

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Event message. Each new event from a particular component gets a new sequence number. The same message might be sent multiple times if (re-)requested. Most events are broadcast, some can be specific to a target component (as receivers keep track of the sequence for missed events, all events need to be broadcast. Thus we use destination_component instead of target_component).

Field Name	Type	Units	Description
destination_component	uint8_t		Component ID
destination_system	uint8_t		System ID
id	uint32_t		Event ID (as defined in the component metadata)
event_time_boot_ms	uint32_t	ms	Timestamp (time since system boot when the event happened).
sequence	uint16_t		Sequence number.
log_levels	uint8_t		Log levels: 4 bits MSB: internal (for logging purposes), 4 bits LSB: external. Levels: Emergency = 0, Alert = 1, Critical = 2, Error = 3, Warning = 4, Notice = 5, Info = 6, Debug = 7, Protocol = 8, Disabled = 9
arguments	uint8_t[40]		Arguments (depend on event ID).

CURRENT_EVENT_SEQUENCE (#411

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Regular broadcast for the current latest event sequence number for a component. This is used to check for dropped events.

Field Name	Type	Values	Description
sequence	uint16_t		Sequence number.
flags	uint8_t	MAV_EVENT_CURRENT_SEQUENCE_FLAGS	Flag bitset.

REQUEST_EVENT (#412

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Request one or more events to be (re-)sent. If first_sequence==last_sequence, only a single event is requested. Note that first_sequence can be larger than last_sequence (because the sequence number can wrap). Each sequence will trigger an EVENT or EVENT_ERROR response.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
first_sequence	uint16_t	First sequence number of the requested event.
last_sequence	uint16_t	Last sequence number of the requested event.

RESPONSE_EVENT_ERROR (#413

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Response to a REQUEST_EVENT in case of an error (e.g. the event is not available anymore).

Field Name	Type	Values	Description
target_system	uint8_t		System ID
target_component	uint8_t		Component ID
sequence	uint16_t		Sequence number.
sequence_oldest_available	uint16_t		Oldest Sequence number that is still available after the sequence set in REQUEST_EVENT.
reason	uint8_t	MAV_EVENT_ERROR_REASON	Error reason.

CANFD_FRAME (#387

)

[Message] **(MAVLink 2)** A forwarded CANFD frame as requested by MAV_CMD_CAN_FORWARD. These are separated from CAN_FRAME as they need different handling (eg. TAO handling)

Field Name	Type	Description
target_system	uint8_t	System ID.
target_component	uint8_t	Component ID.
bus	uint8_t	bus number
len	uint8_t	Frame length
id	uint32_t	Frame ID
data	uint8_t[64]	Frame data

CAN_FILTER_MODIFY (#388

)

[Message] **(MAVLink 2)** Modify the filter of what CAN messages to forward over the mavlink. This can be used to make CAN forwarding work well on low bandwidth links. The filtering is applied on bits 8 to 24 of the CAN id (2nd and 3rd bytes) which corresponds to the DroneCAN message ID for DroneCAN. Filters with more than 16 IDs can be constructed by sending multiple CAN_FILTER_MODIFY messages.

Field Name	Type	Values	Description
target_system	uint8_t		System ID.
target_component	uint8_t		Component ID.
bus	uint8_t		bus number
operation	uint8_t	CAN_FILTER_OP	what operation to perform on the filter list. See CAN_FILTER_OP enum.
num_ids	uint8_t		number of IDs in filter list
ids	uint16_t[16]		filter IDs, length num_ids

WHEEL_DISTANCE (#9000

)

[Message] **(MAVLink 2)** Cumulative distance traveled for each reported wheel.

Field Name	Type	Units	Description
time_usec	uint64_t	us	Timestamp (synced to UNIX time or since system boot).
count	uint8_t		Number of wheels reported.
distance	double[16]	m	Distance reported by individual wheel encoders. Forward rotations increase values, reverse rotations decrease them. Not all wheels will necessarily have wheel encoders; the mapping of encoders to wheel positions must be agreed/understood by the endpoints.

WINCH_STATUS (#9005

)

[Message] **(MAVLink 2)** Winch status.

Field Name	Type	Units	Values	Description
time_usec	uint64_t	us		Timestamp (synced to UNIX time or since system boot).
line_length	float	m		Length of line released. NaN if unknown
speed	float	m/s		Speed line is being released or retracted. Positive values if being released, negative values if being retracted, NaN if unknown
tension	float	kg		Tension on the line. NaN if unknown
voltage	float	V		Voltage of the battery supplying the winch. NaN if unknown
current	float	A		Current draw from the winch. NaN if unknown
temperature	int16_t	degC		Temperature of the motor. INT16_MAX if unknown
status	uint32_t		MAV_WINCH_STATUS_FLAG	Status flags

OPEN_DRONE_ID_BASIC_ID (#12900

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Data for filling the OpenDroneID Basic ID message. This and the below messages are primarily meant for feeding data to/from an OpenDroneID implementation. E.g. https://github.com/opendroneid/opendroneid-core-c. These messages are compatible with the ASTM F3411 Remote ID standard and the ASD-STAN prEN 4709-002 Direct Remote ID standard. Additional information and usage of these messages is documented at https://mavlink.io/en/services/opendroneid.html.

Field Name	Type	Values	Description
target_system	uint8_t		System ID (0 for broadcast).
target_component	uint8_t		Component ID (0 for broadcast).
id_or_mac	uint8_t[20]		Only used for drone ID data received from other UAs. See detailed description at https://mavlink.io/en/services/opendroneid.html.
id_type	uint8_t	MAV_ODID_ID_TYPE	Indicates the format for the uas_id field of this message.
ua_type	uint8_t	MAV_ODID_UA_TYPE	Indicates the type of UA (Unmanned Aircraft).
uas_id	uint8_t[20]		UAS (Unmanned Aircraft System) ID following the format specified by id_type. Shall be filled with nulls in the unused portion of the field.

OPEN_DRONE_ID_LOCATION (#12901

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Data for filling the OpenDroneID Location message. The float data types are 32-bit IEEE 754. The Location message provides the location, altitude, direction and speed of the aircraft.

Field Name	Type	Units	Values	Description
target_system	uint8_t			System ID (0 for broadcast).
target_component	uint8_t			Component ID (0 for broadcast).
id_or_mac	uint8_t[20]			Only used for drone ID data received from other UAs. See detailed description at https://mavlink.io/en/services/opendroneid.html.
status	uint8_t		MAV_ODID_STATUS	Indicates whether the unmanned aircraft is on the ground or in the air.
direction	uint16_t	cdeg		Direction over ground (not heading, but direction of movement) measured clockwise from true North: 0 - 35999 centi-degrees. If unknown: 36100 centi-degrees.
speed_horizontal	uint16_t	cm/s		Ground speed. Positive only. If unknown: 25500 cm/s. If speed is larger than 25425 cm/s, use 25425 cm/s.
speed_vertical	int16_t	cm/s		The vertical speed. Up is positive. If unknown: 6300 cm/s. If speed is larger than 6200 cm/s, use 6200 cm/s. If lower than -6200 cm/s, use -6200 cm/s.
latitude	int32_t	degE7		Current latitude of the unmanned aircraft. If unknown: 0 (both Lat/Lon).
longitude	int32_t	degE7		Current longitude of the unmanned aircraft. If unknown: 0 (both Lat/Lon).
altitude_barometric	float	m		The altitude calculated from the barometric pressue. Reference is against 29.92inHg or 1013.2mb. If unknown: -1000 m.
altitude_geodetic	float	m		The geodetic altitude as defined by WGS84. If unknown: -1000 m.
height_reference	uint8_t		MAV_ODID_HEIGHT_REF	Indicates the reference point for the height field.
height	float	m		The current height of the unmanned aircraft above the take-off location or the ground as indicated by height_reference. If unknown: -1000 m.
horizontal_accuracy	uint8_t		MAV_ODID_HOR_ACC	The accuracy of the horizontal position.
vertical_accuracy	uint8_t		MAV_ODID_VER_ACC	The accuracy of the vertical position.
barometer_accuracy	uint8_t		MAV_ODID_VER_ACC	The accuracy of the barometric altitude.
speed_accuracy	uint8_t		MAV_ODID_SPEED_ACC	The accuracy of the horizontal and vertical speed.
timestamp	float	s		Seconds after the full hour with reference to UTC time. Typically the GPS outputs a time-of-week value in milliseconds. First convert that to UTC and then convert for this field using ((float) (time_week_ms % (60*60*1000))) / 1000. If unknown: 0xFFFF.
timestamp_accuracy	uint8_t		MAV_ODID_TIME_ACC	The accuracy of the timestamps.

OPEN_DRONE_ID_AUTHENTICATION (#12902

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Data for filling the OpenDroneID Authentication message. The Authentication Message defines a field that can provide a means of authenticity for the identity of the UAS (Unmanned Aircraft System). The Authentication message can have two different formats. For data page 0, the fields PageCount, Length and TimeStamp are present and AuthData is only 17 bytes. For data page 1 through 15, PageCount, Length and TimeStamp are not present and the size of AuthData is 23 bytes.

Field Name	Type	Units	Values	Description
target_system	uint8_t			System ID (0 for broadcast).
target_component	uint8_t			Component ID (0 for broadcast).
id_or_mac	uint8_t[20]			Only used for drone ID data received from other UAs. See detailed description at https://mavlink.io/en/services/opendroneid.html.
authentication_type	uint8_t		MAV_ODID_AUTH_TYPE	Indicates the type of authentication.
data_page	uint8_t			Allowed range is 0 - 15.
last_page_index	uint8_t			This field is only present for page 0. Allowed range is 0 - 15. See the description of struct ODID_Auth_data at https://github.com/opendroneid/opendroneid-core-c/blob/master/libopendroneid/opendroneid.h.
length	uint8_t	bytes		This field is only present for page 0. Total bytes of authentication_data from all data pages. See the description of struct ODID_Auth_data at https://github.com/opendroneid/opendroneid-core-c/blob/master/libopendroneid/opendroneid.h.
timestamp	uint32_t	s		This field is only present for page 0. 32 bit Unix Timestamp in seconds since 00:00:00 01/01/2019.
authentication_data	uint8_t[23]			Opaque authentication data. For page 0, the size is only 17 bytes. For other pages, the size is 23 bytes. Shall be filled with nulls in the unused portion of the field.

OPEN_DRONE_ID_SELF_ID (#12903

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Data for filling the OpenDroneID Self ID message. The Self ID Message is an opportunity for the operator to (optionally) declare their identity and purpose of the flight. This message can provide additional information that could reduce the threat profile of a UA (Unmanned Aircraft) flying in a particular area or manner. This message can also be used to provide optional additional clarification in an emergency/remote ID system failure situation.

Field Name	Type	Values	Description
target_system	uint8_t		System ID (0 for broadcast).
target_component	uint8_t		Component ID (0 for broadcast).
id_or_mac	uint8_t[20]		Only used for drone ID data received from other UAs. See detailed description at https://mavlink.io/en/services/opendroneid.html.
description_type	uint8_t	MAV_ODID_DESC_TYPE	Indicates the type of the description field.
description	char[23]		Text description or numeric value expressed as ASCII characters. Shall be filled with nulls in the unused portion of the field.

OPEN_DRONE_ID_SYSTEM (#12904

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Data for filling the OpenDroneID System message. The System Message contains general system information including the operator location/altitude and possible aircraft group and/or category/class information.

Field Name	Type	Units	Values	Description
target_system	uint8_t			System ID (0 for broadcast).
target_component	uint8_t			Component ID (0 for broadcast).
id_or_mac	uint8_t[20]			Only used for drone ID data received from other UAs. See detailed description at https://mavlink.io/en/services/opendroneid.html.
operator_location_type	uint8_t		MAV_ODID_OPERATOR_LOCATION_TYPE	Specifies the operator location type.
classification_type	uint8_t		MAV_ODID_CLASSIFICATION_TYPE	Specifies the classification type of the UA.
operator_latitude	int32_t	degE7		Latitude of the operator. If unknown: 0 (both Lat/Lon).
operator_longitude	int32_t	degE7		Longitude of the operator. If unknown: 0 (both Lat/Lon).
area_count	uint16_t			Number of aircraft in the area, group or formation (default 1).
area_radius	uint16_t	m		Radius of the cylindrical area of the group or formation (default 0).
area_ceiling	float	m		Area Operations Ceiling relative to WGS84. If unknown: -1000 m.
area_floor	float	m		Area Operations Floor relative to WGS84. If unknown: -1000 m.
category_eu	uint8_t		MAV_ODID_CATEGORY_EU	When classification_type is MAV_ODID_CLASSIFICATION_TYPE_EU, specifies the category of the UA.
class_eu	uint8_t		MAV_ODID_CLASS_EU	When classification_type is MAV_ODID_CLASSIFICATION_TYPE_EU, specifies the class of the UA.
operator_altitude_geo	float	m		Geodetic altitude of the operator relative to WGS84. If unknown: -1000 m.
timestamp	uint32_t	s		32 bit Unix Timestamp in seconds since 00:00:00 01/01/2019.

OPEN_DRONE_ID_OPERATOR_ID (#12905

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** Data for filling the OpenDroneID Operator ID message, which contains the CAA (Civil Aviation Authority) issued operator ID.

Field Name	Type	Values	Description
target_system	uint8_t		System ID (0 for broadcast).
target_component	uint8_t		Component ID (0 for broadcast).
id_or_mac	uint8_t[20]		Only used for drone ID data received from other UAs. See detailed description at https://mavlink.io/en/services/opendroneid.html.
operator_id_type	uint8_t	MAV_ODID_OPERATOR_ID_TYPE	Indicates the type of the operator_id field.
operator_id	char[20]		Text description or numeric value expressed as ASCII characters. Shall be filled with nulls in the unused portion of the field.

OPEN_DRONE_ID_MESSAGE_PACK (#12915

)

WORK IN PROGRESS: Do not use in stable production environments (it may change).

[Message] **(MAVLink 2)** An OpenDroneID message pack is a container for multiple encoded OpenDroneID messages (i.e. not in the format given for the above message descriptions but after encoding into the compressed OpenDroneID byte format). Used e.g. when transmitting on Bluetooth 5.0 Long Range/Extended Advertising or on WiFi Neighbor Aware Networking or on WiFi Beacon.

Field Name	Type	Units	Description
target_system	uint8_t		System ID (0 for broadcast).
target_component	uint8_t		Component ID (0 for broadcast).
id_or_mac	uint8_t[20]		Only used for drone ID data received from other UAs. See detailed description at https://mavlink.io/en/services/opendroneid.html.
single_message_size	uint8_t	bytes	This field must currently always be equal to 25 (bytes), since all encoded OpenDroneID messages are specified to have this length.
msg_pack_size	uint8_t		Number of encoded messages in the pack (not the number of bytes). Allowed range is 1 - 9.
messages	uint8_t[225]		Concatenation of encoded OpenDroneID messages. Shall be filled with nulls in the unused portion of the field.

HYGROMETER_SENSOR (#12920

)

[Message] **(MAVLink 2)** Temperature and humidity from hygrometer.

Field Name	Type	Units	Description
id	uint8_t		Hygrometer ID
temperature	int16_t	cdegC	Temperature
humidity	uint16_t	c%	Humidity