MAVLINK Common Message Set

These messages define the common message set, which is the reference message set implemented by most ground control stations and autopilots.

MAVLink Protocol Version

This file has protocol version: 3. The version numbers range from 1-255.

0

MAVLink Type Enumerations

MAV_AUTOPILOT

Micro air vehicle / autopilot classes. This identifies the individual model.

CMD ID	Field Name	Description
0	**MAV_AUTOPILOT_GENERIC**	Generic autopilot, full support for everything
1	**MAV_AUTOPILOT_RESERVED**	Reserved for future use.
2	**MAV_AUTOPILOT_SLUGS**	SLUGS autopilot, http://slugsuav.soe.ucsc.edu
3	**MAV_AUTOPILOT_ARDUPILOTMEGA**	ArduPilotMega / ArduCopter, http://diydrones.com
4	**MAV_AUTOPILOT_OPENPILOT**	OpenPilot, http://openpilot.org
5	**MAV_AUTOPILOT_GENERIC_WAYPOINTS_ONLY**	Generic autopilot only supporting simple waypoints
6	**MAV_AUTOPILOT_GENERIC_WAYPOINTS_AND_SIMPLE_NAVIGATION_ONLY**	Generic autopilot supporting waypoints and other simple navigation commands
7	**MAV_AUTOPILOT_GENERIC_MISSION_FULL**	Generic autopilot supporting the full mission command set
8	**MAV_AUTOPILOT_INVALID**	No valid autopilot, e.g. a GCS or other MAVLink

		component
9	**MAV_AUTOPILOT_PPZ**	PPZ UAV - http://nongnu.org/paparazzi
10	**MAV_AUTOPILOT_UDB**	UAV Dev Board
11	**MAV_AUTOPILOT_FP**	FlexiPilot
12	**MAV_AUTOPILOT_PX4**	PX4 Autopilot - http://pixhawk.ethz.ch/px4/
13	**MAV_AUTOPILOT_SMACCMPILOT**	SMACCMPilot - http://smaccmpilot.org
14	**MAV_AUTOPILOT_AUTOQUAD**	AutoQuad -- http://autoquad.org
15	**MAV_AUTOPILOT_ARMAZILA**	Armazila -- http://armazila.com
16	**MAV_AUTOPILOT_AEROB**	Aerob -- http://aerob.ru
17	**MAV_AUTOPILOT_ASLUAV**	ASLUAV autopilot -- http://www.asl.ethz.ch

MAV_TYPE

CMD ID	Field Name	Description
0	**MAV_TYPE_GENERIC**	Generic micro air vehicle.
1	**MAV_TYPE_FIXED_WING**	Fixed wing aircraft.
2	**MAV_TYPE_QUADROTOR**	Quadrotor
3	**MAV_TYPE_COAXIAL**	Coaxial helicopter
4	**MAV_TYPE_HELICOPTER**	Normal helicopter with tail rotor.
5	**MAV_TYPE_ANTENNA_TRACKER**	Ground installation
6	**MAV_TYPE_GCS**	Operator control unit / ground control station
7	**MAV_TYPE_AIRSHIP**	Airship, controlled
8	**MAV_TYPE_FREE_BALLOON**	Free balloon, uncontrolled

9	**MAV_TYPE_ROCKET**	Rocket
10	**MAV_TYPE_GROUND_ROVER**	Ground rover
11	**MAV_TYPE_SURFACE_BOAT**	Surface vessel, boat, ship
12	**MAV_TYPE_SUBMARINE**	Submarine
13	**MAV_TYPE_HEXAROTOR**	Hexarotor
14	**MAV_TYPE_OCTOROTOR**	Octorotor
15	**MAV_TYPE_TRICOPTER**	Octorotor
16	**MAV_TYPE_FLAPPING_WING**	Flapping wing
17	**MAV_TYPE_KITE**	Flapping wing
18	**MAV_TYPE_ONBOARD_CONTROLLER**	Onboard companion controller
19	**MAV_TYPE_VTOL_DUOROTOR**	Two-rotor VTOL using control surfaces in vertical operation in addition. Tailsitter.
20	**MAV_TYPE_VTOL_QUADROTOR**	Quad-rotor VTOL using a V-shaped quad config in vertical operation. Tailsitter.
21	**MAV_TYPE_VTOL_TILTROTOR**	Tiltrotor VTOL
22	**MAV_TYPE_VTOL_RESERVED2**	VTOL reserved 2
23	**MAV_TYPE_VTOL_RESERVED3**	VTOL reserved 3
24	**MAV_TYPE_VTOL_RESERVED4**	VTOL reserved 4
25	**MAV_TYPE_VTOL_RESERVED5**	VTOL reserved 5
26	**MAV_TYPE_GIMBAL**	Onboard gimbal
27	**MAV_TYPE_ADSB**	Onboard ADSB peripheral

FIRMWARE_VERSION_TYPE

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.

CMD ID	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

MAV_MODE_FLAG

These flags encode the MAV mode.

CMD ID	Field Name	Description
128	**MAV_MODE_FLAG_SAFETY_ARMED**	0b10000000 MAV safety set to armed. Motors are enabled / running / can start. Ready to fly.
64	**MAV_MODE_FLAG_MANUAL_INPUT_ENABLED**	0b01000000 remote control input is enabled.
32	**MAV_MODE_FLAG_HIL_ENABLED**	0b00100000 hardware in the loop simulation. All motors / actuators are blocked, but internal software is full operational.
16	**MAV_MODE_FLAG_STABILIZE_ENABLED**	0b00010000 system stabilizes electronically its attitude (and optionally position). It needs however further control inputs to move around.
8	**MAV_MODE_FLAG_GUIDED_ENABLED**	0b00001000 guided mode enabled, system flies MISSIONs / mission items.
4	**MAV_MODE_FLAG_AUTO_ENABLED**	0b00000100 autonomous mode enabled, system finds its own goal positions. Guided flag can be set or not, depends on the actual implementation.
2	**MAV_MODE_FLAG_TEST_ENABLED**	0b00000010 system has a test mode enabled. This flag is intended for temporary system tests and should not be used for stable implementations.
1	**MAV_MODE_FLAG_CUSTOM_MODE_ENABLED**	0b00000001 Reserved for future use.

MAV_MODE_FLAG_DECODE_POSITION

These values encode the bit positions of the decode position. These values can be used to read the value of a flag bit by combining the base_mode variable with AND with the flag position value. The result will be either 0 or 1, depending on if the flag is set or not.

CMD ID	Field Name	Description
128	**MAV_MODE_FLAG_DECODE_POSITION_SAFETY**	First bit: 10000000
64	**MAV_MODE_FLAG_DECODE_POSITION_MANUAL**	Second bit: 01000000
32	**MAV_MODE_FLAG_DECODE_POSITION_HIL**	Third bit: 00100000
16	**MAV_MODE_FLAG_DECODE_POSITION_STABILIZE**	Fourth bit: 00010000
8	**MAV_MODE_FLAG_DECODE_POSITION_GUIDED**	Fifth bit: 00001000
4	**MAV_MODE_FLAG_DECODE_POSITION_AUTO**	Sixt bit: 00000100
2	**MAV_MODE_FLAG_DECODE_POSITION_TEST**	Seventh bit: 00000010
1	**MAV_MODE_FLAG_DECODE_POSITION_CUSTOM_MODE**	Eighth bit: 00000001

MAV_GOTO

Override command, pauses current mission execution and moves immediately to a position

CMD ID	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

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.

CMD ID	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 MISSIONs)
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 MISSIONs)
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_STATE

CMD ID	Field Name	Description
0	**MAV_STATE_UNINIT**	Uninitialized system, state is unknown.
	MAV_STATE_BOOT	System is booting up.
	MAV_STATE_CALIBRATING	System is calibrating and not flight-ready.
	MAV_STATE_STANDBY	System is grounded and on standby. It can be launched any time.
	MAV_STATE_ACTIVE	System is active and might be already airborne. Motors are engaged.
	MAV_STATE_CRITICAL	System is in a non-normal flight mode. It can however still

		navigate.
	MAV_STATE_EMERGENCY	System is in a non-normal flight mode. It lost control over parts or over the whole airframe. It is in mayday and going down.
	MAV_STATE_POWEROFF	System just initialized its power-down sequence, will shut down now.

MAV_COMPONENT

CMD ID	Field Name	Description
0	**MAV_COMP_ID_ALL**	
220	**MAV_COMP_ID_GPS**	
190	**MAV_COMP_ID_MISSIONPLANNER**	
195	**MAV_COMP_ID_PATHPLANNER**	
180	**MAV_COMP_ID_MAPPER**	
100	**MAV_COMP_ID_CAMERA**	
200	**MAV_COMP_ID_IMU**	
201	**MAV_COMP_ID_IMU_2**	
202	**MAV_COMP_ID_IMU_3**	
240	**MAV_COMP_ID_UDP_BRIDGE**	
241	**MAV_COMP_ID_UART_BRIDGE**	
250	**MAV_COMP_ID_SYSTEM_CONTROL**	
140	**MAV_COMP_ID_SERVO1**	
141	**MAV_COMP_ID_SERVO2**	
142	**MAV_COMP_ID_SERVO3**	
143	**MAV_COMP_ID_SERVO4**	
144	**MAV_COMP_ID_SERVO5**	
145	**MAV_COMP_ID_SERVO6**	

146	**MAV_COMP_ID_SERVO7**	
147	**MAV_COMP_ID_SERVO8**	
148	**MAV_COMP_ID_SERVO9**	
149	**MAV_COMP_ID_SERVO10**	
150	**MAV_COMP_ID_SERVO11**	
151	**MAV_COMP_ID_SERVO12**	
152	**MAV_COMP_ID_SERVO13**	
153	**MAV_COMP_ID_SERVO14**	
154	**MAV_COMP_ID_GIMBAL**	
155	**MAV_COMP_ID_LOG**	
156	**MAV_COMP_ID_ADSB**	
157	**MAV_COMP_ID_OSD**	On Screen Display (OSD) devices for video links
158	**MAV_COMP_ID_PERIPHERAL**	Generic autopilot peripheral component ID. Meant for devices that do not implement the parameter sub-protocol
159	**MAV_COMP_ID_QX1_GIMBAL**	

MAV_SYS_STATUS_SENSOR

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

CMD ID	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

MAV_FRAME

CMD ID	Field Name	Description
0	**MAV_FRAME_GLOBAL**	Global coordinate frame, WGS84 coordinate

		system. First value / x: latitude, second value / y: longitude, third value / z: positive altitude over mean sea level (MSL)
1	**MAV_FRAME_LOCAL_NED**	Local coordinate frame, Z-up (x: north, y: east, z: down).
2	**MAV_FRAME_MISSION**	NOT a coordinate frame, indicates a mission command.
3	**MAV_FRAME_GLOBAL_RELATIVE_ALT**	Global coordinate frame, WGS84 coordinate system, relative altitude over ground with respect 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 location.
4	**MAV_FRAME_LOCAL_ENU**	Local coordinate frame, Z-down (x: east, y: north, z: up)
5	**MAV_FRAME_GLOBAL_INT**	Global coordinate frame, WGS84 coordinate system. First value / x: latitude in degrees*1.0e-7, second value / y: longitude in degrees*1.0e-7, third value / z: positive altitude over mean sea level (MSL)
6	**MAV_FRAME_GLOBAL_RELATIVE_ALT_INT**	Global coordinate frame, WGS84 coordinate system, relative altitude over ground with respect to the home position. First value / x: latitude in degrees*10e-7, second value / y: longitude in degrees*10e-7, third value / z: positive altitude with 0 being at the altitude of the home location.
7	**MAV_FRAME_LOCAL_OFFSET_NED**	Offset to the current local frame. Anything expressed in this frame should be added to the current local frame position.
8	**MAV_FRAME_BODY_NED**	Setpoint in body NED frame. This makes sense if all position control is externalized - e.g. useful to command 2 m/s^2 acceleration to the right.
9	**MAV_FRAME_BODY_OFFSET_NED**	Offset in body NED frame. This makes sense if adding setpoints to the current flight path, to avoid an obstacle - e.g. useful to command 2 m/s^2 acceleration to the east.
10	**MAV_FRAME_GLOBAL_TERRAIN_ALT**	Global coordinate frame with above terrain level altitude. WGS84 coordinate system, relative altitude over terrain with respect to 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 coordinate frame with above terrain level altitude. WGS84 coordinate system, relative altitude over terrain with respect to the waypoint

		coordinate. First value / x: latitude in degrees*10e-7, second value / y: longitude in degrees*10e-7, third value / z: positive altitude in meters with 0 being at ground level in terrain model.

MAVLINK_DATA_STREAM_TYPE

CMD ID	Field Name	Description
	MAVLINK_DATA_STREAM_IMG_JPEG	
	MAVLINK_DATA_STREAM_IMG_BMP	
	MAVLINK_DATA_STREAM_IMG_RAW8U	
	MAVLINK_DATA_STREAM_IMG_RAW32U	
	MAVLINK_DATA_STREAM_IMG_PGM	
	MAVLINK_DATA_STREAM_IMG_PNG	

FENCE_ACTION

CMD ID	Field Name	Description
0	**FENCE_ACTION_NONE**	Disable fenced mode
1	**FENCE_ACTION_GUIDED**	Switched to guided mode to return point (fence point 0)
2	**FENCE_ACTION_REPORT**	Report fence breach, but don't take action
3	**FENCE_ACTION_GUIDED_THR_PASS**	Switched to guided mode to return point (fence point 0) with manual throttle control
4	**FENCE_ACTION_RTL**	Switch to RTL (return to launch) mode and head for the return point.

FENCE_BREACH

CMD ID	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

MAV_MOUNT_MODE

Enumeration of possible mount operation modes

CMD ID	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

MAV_CMD

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.

CMD ID	Field Name	Description
16	**MAV_CMD_NAV_WAYPOINT**	Navigate to MISSION.
	Mission Param #1	Hold time in decimal seconds. (ignored by fixed wing, time to stay at MISSION for rotary wing)
	Mission Param #2	Acceptance radius in meters (if the sphere with this radius is hit, the MISSION counts as reached)
	Mission Param #3	0 to pass through the WP, if > 0 radius in meters to pass by WP. Positive value for clockwise orbit, negative value for counter-clockwise orbit. Allows trajectory control.
	Mission Param #4	Desired yaw angle at MISSION (rotary wing)
	Mission Param #5	Latitude

	Mission Param #6	Longitude
	Mission Param #7	Altitude
17	**MAV_CMD_NAV_LOITER_UNLIM**	Loiter around this MISSION an unlimited amount of time
	Mission Param #1	Empty
	Mission Param #2	Empty
	Mission Param #3	Radius around MISSION, in meters. If positive loiter clockwise, else counter-clockwise
	Mission Param #4	Desired yaw angle.
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Altitude
18	**MAV_CMD_NAV_LOITER_TURNS**	Loiter around this MISSION for X turns
	Mission Param #1	Turns
	Mission Param #2	Empty
	Mission Param #3	Radius around MISSION, in meters. If positive loiter clockwise, else counter-clockwise
	Mission Param #4	Forward moving aircraft this sets exit xtrack location: 0 for center of loiter wp, 1 for exit location. Else, this is desired yaw angle
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Altitude
19	**MAV_CMD_NAV_LOITER_TIME**	Loiter around this MISSION for X seconds
	Mission Param #1	Seconds (decimal)
	Mission Param #2	Empty
	Mission Param #3	Radius around MISSION, in meters. If positive loiter clockwise, else counter-clockwise
	Mission Param #4	Forward moving aircraft this sets exit xtrack location: 0 for center of loiter wp, 1 for exit location. Else, this is desired yaw angle
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Altitude
20	**MAV_CMD_NAV_RETURN_TO_LAUNCH**	Return to launch location
	Mission Param #1	Empty
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
21	**MAV_CMD_NAV_LAND**	Land at location
	Mission Param #1	Abort Alt
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Desired yaw angle
	Mission Param #5	Latitude
	Mission Param #6	Longitude

	Mission Param #7	Altitude
22	**MAV_CMD_NAV_TAKEOFF**	Takeoff from ground / hand
	Mission Param #1	Minimum pitch (if airspeed sensor present), desired pitch without sensor
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Yaw angle (if magnetometer present), ignored without magnetometer
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Altitude
23	**MAV_CMD_NAV_LAND_LOCAL**	Land at local position (local frame only)
	Mission Param #1	Landing target number (if available)
	Mission Param #2	Maximum accepted offset from desired landing position [m] - 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
	Mission Param #3	Landing descend rate [ms^-1]
	Mission Param #4	Desired yaw angle [rad]
	Mission Param #5	Y-axis position [m]
	Mission Param #6	X-axis position [m]
	Mission Param #7	Z-axis / ground level position [m]
24	**MAV_CMD_NAV_TAKEOFF_LOCAL**	Takeoff from local position (local frame only)
	Mission Param #1	Minimum pitch (if airspeed sensor present), desired pitch without sensor [rad]
	Mission Param #2	Empty
	Mission Param #3	Takeoff ascend rate [ms^-1]
	Mission Param #4	Yaw angle [rad] (if magnetometer or another yaw estimation source present), ignored without one of these
	Mission Param #5	Y-axis position [m]
	Mission Param #6	X-axis position [m]
	Mission Param #7	Z-axis position [m]
25	**MAV_CMD_NAV_FOLLOW**	Vehicle following, i.e. this waypoint represents the position of a moving vehicle
	Mission Param #1	Following logic to use (e.g. loitering or sinusoidal following) - depends on specific autopilot implementation
	Mission Param #2	Ground speed of vehicle to be followed
	Mission Param #3	Radius around MISSION, in meters. If positive loiter clockwise, else counter-clockwise
	Mission Param #4	Desired yaw angle.
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Altitude
30	**MAV_CMD_NAV_CONTINUE_AND_CHANGE_ALT**	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

		desired altitude is reached.
	Mission Param #1	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.
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Desired altitude in meters
31	**MAV_CMD_NAV_LOITER_TO_ALT**	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.
	Mission Param #1	Heading Required (0 = False)
	Mission Param #2	Radius in meters. If positive loiter clockwise, negative counter-clockwise, 0 means no change to standard loiter.
	Mission Param #3	Empty
	Mission Param #4	Forward moving aircraft this sets exit xtrack location: 0 for center of loiter wp, 1 for exit location
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Altitude
32	**MAV_CMD_DO_FOLLOW**	Being following a target
	Mission Param #1	System ID (the system ID of the FOLLOW_TARGET beacon). Send 0 to disable follow-me and return to the default position hold mode
	Mission Param #2	RESERVED
	Mission Param #3	RESERVED
	Mission Param #4	altitude flag: 0: Keep current altitude, 1: keep altitude difference to target, 2: go to a fixed altitude above home
	Mission Param #5	altitude
	Mission Param #6	RESERVED
	Mission Param #7	TTL in seconds in which the MAV should go to the default position hold mode after a message rx timeout
33	**MAV_CMD_DO_FOLLOW_REPOSITION**	Reposition the MAV after a follow target command has been sent
	Mission Param #1	Camera q1 (where 0 is on the ray from the camera to the tracking device)
	Mission Param #2	Camera q2
	Mission Param #3	Camera q3
	Mission Param #4	Camera q4

	Mission Param #5	altitude offset from target (m)
	Mission Param #6	X offset from target (m)
	Mission Param #7	Y offset from target (m)
80	**MAV_CMD_NAV_ROI**	Sets the region of interest (ROI) for a sensor set or the vehicle itself. This can then be used by the vehicles control system to control the vehicle attitude and the attitude of various sensors such as cameras.
	Mission Param #1	Region of intereset mode. (see MAV_ROI enum)
	Mission Param #2	MISSION index/ target ID. (see MAV_ROI enum)
	Mission Param #3	ROI index (allows a vehicle to manage multiple ROI's)
	Mission Param #4	Empty
	Mission Param #5	x the location of the fixed ROI (see MAV_FRAME)
	Mission Param #6	y
	Mission Param #7	z
81	**MAV_CMD_NAV_PATHPLANNING**	Control autonomous path planning on the MAV.
	Mission Param #1	0: Disable local obstacle avoidance / local path planning (without resetting map), 1: Enable local path planning, 2: Enable and reset local path planning
	Mission Param #2	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
	Mission Param #3	Empty
	Mission Param #4	Yaw angle at goal, in compass degrees, [0..360]
	Mission Param #5	Latitude/X of goal
	Mission Param #6	Longitude/Y of goal
	Mission Param #7	Altitude/Z of goal
82	**MAV_CMD_NAV_SPLINE_WAYPOINT**	Navigate to MISSION using a spline path.
	Mission Param #1	Hold time in decimal seconds. (ignored by fixed wing, time to stay at MISSION for rotary wing)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Latitude/X of goal
	Mission Param #6	Longitude/Y of goal
	Mission Param #7	Altitude/Z of goal
84	**MAV_CMD_NAV_VTOL_TAKEOFF**	Takeoff from ground using VTOL mode
	Mission Param #1	Empty
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Yaw angle in degrees
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Altitude
85	**MAV_CMD_NAV_VTOL_LAND**	Land using VTOL mode
	Mission Param #1	Empty

	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Yaw angle in degrees
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Altitude
92	**MAV_CMD_NAV_GUIDED_ENABLE**	hand control over to an external controller
	Mission Param #1	On / Off (> 0.5f on)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
93	**MAV_CMD_NAV_DELAY**	Delay the next navigation command a number of seconds or until a specified time
	Mission Param #1	Delay in seconds (decimal, -1 to enable time-of-day fields)
	Mission Param #2	hour (24h format, UTC, -1 to ignore)
	Mission Param #3	minute (24h format, UTC, -1 to ignore)
	Mission Param #4	second (24h format, UTC)
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
95	**MAV_CMD_NAV_LAST**	NOP - This command is only used to mark the upper limit of the NAV/ACTION commands in the enumeration
	Mission Param #1	Empty
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
112	**MAV_CMD_CONDITION_DELAY**	Delay mission state machine.
	Mission Param #1	Delay in seconds (decimal)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
113	**MAV_CMD_CONDITION_CHANGE_ALT**	Ascend/descend at rate. Delay mission state machine until desired altitude reached.
	Mission Param #1	Descent / Ascend rate (m/s)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty

	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Finish Altitude
114	**MAV_CMD_CONDITION_DISTANCE**	Delay mission state machine until within desired distance of next NAV point.
	Mission Param #1	Distance (meters)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
115	**MAV_CMD_CONDITION_YAW**	Reach a certain target angle.
	Mission Param #1	target angle: [0-360], 0 is north
	Mission Param #2	speed during yaw change:[deg per second]
	Mission Param #3	direction: negative: counter clockwise, positive: clockwise [-1,1]
	Mission Param #4	relative offset or absolute angle: [1,0]
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
159	**MAV_CMD_CONDITION_LAST**	NOP - This command is only used to mark the upper limit of the CONDITION commands in the enumeration
	Mission Param #1	Empty
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
176	**MAV_CMD_DO_SET_MODE**	Set system mode.
	Mission Param #1	Mode, as defined by ENUM MAV_MODE
	Mission Param #2	Custom mode - this is system specific, please refer to the individual autopilot specifications for details.
	Mission Param #3	Custom sub mode - this is system specific, please refer to the individual autopilot specifications for details.
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
177	**MAV_CMD_DO_JUMP**	Jump to the desired command in the mission list. Repeat this action only the specified number of times
	Mission Param #1	Sequence number
	Mission Param #2	Repeat count
	Mission Param #3	Empty

	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
178	**MAV_CMD_DO_CHANGE_SPEED**	Change speed and/or throttle set points.
	Mission Param #1	Speed type (0=Airspeed, 1=Ground Speed)
	Mission Param #2	Speed (m/s, -1 indicates no change)
	Mission Param #3	Throttle (Percent, -1 indicates no change)
	Mission Param #4	absolute or relative [0,1]
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
179	**MAV_CMD_DO_SET_HOME**	Changes the home location either to the current location or a specified location.
	Mission Param #1	Use current (1=use current location, 0=use specified location)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Altitude
180	**MAV_CMD_DO_SET_PARAMETER**	Set a system parameter. Caution! Use of this command requires knowledge of the numeric enumeration value of the parameter.
	Mission Param #1	Parameter number
	Mission Param #2	Parameter value
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
181	**MAV_CMD_DO_SET_RELAY**	Set a relay to a condition.
	Mission Param #1	Relay number
	Mission Param #2	Setting (1=on, 0=off, others possible depending on system hardware)
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
182	**MAV_CMD_DO_REPEAT_RELAY**	Cycle a relay on and off for a desired number of cyles with a desired period.
	Mission Param #1	Relay number
	Mission Param #2	Cycle count
	Mission Param #3	Cycle time (seconds, decimal)
	Mission Param #4	Empty
	Mission Param #5	Empty

	Mission Param #6	Empty
	Mission Param #7	Empty
183	**MAV_CMD_DO_SET_SERVO**	Set a servo to a desired PWM value.
	Mission Param #1	Servo number
	Mission Param #2	PWM (microseconds, 1000 to 2000 typical)
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
184	**MAV_CMD_DO_REPEAT_SERVO**	Cycle a between its nominal setting and a desired PWM for a desired number of cycles with a desired period.
	Mission Param #1	Servo number
	Mission Param #2	PWM (microseconds, 1000 to 2000 typical)
	Mission Param #3	Cycle count
	Mission Param #4	Cycle time (seconds)
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
185	**MAV_CMD_DO_FLIGHTTERMINATION**	Terminate flight immediately
	Mission Param #1	Flight termination activated if > 0.5
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
186	**MAV_CMD_DO_CHANGE_ALTITUDE**	Change altitude set point.
	Mission Param #1	Altitude in meters
	Mission Param #2	Mav frame of new altitude (see MAV_FRAME)
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
189	**MAV_CMD_DO_LAND_START**	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 0/0 if not needed. If specified then it will be used to help find the closest landing sequence.
	Mission Param #1	Empty
	Mission Param #2	Empty
	Mission Param #3	Empty

	Mission Param #4	Empty
	Mission Param #5	Latitude
	Mission Param #6	Longitude
	Mission Param #7	Empty
190	**MAV_CMD_DO_RALLY_LAND**	Mission command to perform a landing from a rally point.
	Mission Param #1	Break altitude (meters)
	Mission Param #2	Landing speed (m/s)
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
191	**MAV_CMD_DO_GO_AROUND**	Mission command to safely abort an autonmous landing.
	Mission Param #1	Altitude (meters)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
192	**MAV_CMD_DO_REPOSITION**	Reposition the vehicle to a specific WGS84 global position.
	Mission Param #1	Ground speed, less than 0 (-1) for default
	Mission Param #2	Bitmask of option flags, see the MAV_DO_REPOSITION_FLAGS enum.
	Mission Param #3	Reserved
	Mission Param #4	Yaw heading, NaN for unchanged. For planes indicates loiter direction (0: clockwise, 1: counter clockwise)
	Mission Param #5	Latitude (deg * 1E7)
	Mission Param #6	Longitude (deg * 1E7)
	Mission Param #7	Altitude (meters)
193	**MAV_CMD_DO_PAUSE_CONTINUE**	If in a GPS controlled position mode, hold the current position or continue.
	Mission Param #1	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.
	Mission Param #2	Reserved
	Mission Param #3	Reserved
	Mission Param #4	Reserved
	Mission Param #5	Reserved
	Mission Param #6	Reserved
	Mission Param #7	Reserved
194	**MAV_CMD_DO_SET_REVERSE**	Set moving direction to forward or reverse.
	Mission Param #1	Direction (0=Forward, 1=Reverse)

	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
200	**MAV_CMD_DO_CONTROL_VIDEO**	Control onboard camera system.
	Mission Param #1	Camera ID (-1 for all)
	Mission Param #2	Transmission: 0: disabled, 1: enabled compressed, 2: enabled raw
	Mission Param #3	Transmission mode: 0: video stream, >0: single images every n seconds (decimal)
	Mission Param #4	Recording: 0: disabled, 1: enabled compressed, 2: enabled raw
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
201	**MAV_CMD_DO_SET_ROI**	Sets the region of interest (ROI) for a sensor set or the vehicle itself. This can then be used by the vehicles control system to control the vehicle attitude and the attitude of various sensors such as cameras.
	Mission Param #1	Region of intereset mode. (see MAV_ROI enum)
	Mission Param #2	MISSION index/ target ID. (see MAV_ROI enum)
	Mission Param #3	ROI index (allows a vehicle to manage multiple ROI's)
	Mission Param #4	Empty
	Mission Param #5	x the location of the fixed ROI (see MAV_FRAME)
	Mission Param #6	y
	Mission Param #7	z
202	**MAV_CMD_DO_DIGICAM_CONFIGURE**	Mission command to configure an on-board camera controller system.
	Mission Param #1	Modes: P, TV, AV, M, Etc
	Mission Param #2	Shutter speed: Divisor number for one second
	Mission Param #3	Aperture: F stop number
	Mission Param #4	ISO number e.g. 80, 100, 200, Etc
	Mission Param #5	Exposure type enumerator
	Mission Param #6	Command Identity
	Mission Param #7	Main engine cut-off time before camera trigger in seconds/10 (0 means no cut-off)
203	**MAV_CMD_DO_DIGICAM_CONTROL**	Mission command to control an on-board camera controller system.
	Mission Param #1	Session control e.g. show/hide lens
	Mission Param #2	Zoom's absolute position
	Mission Param #3	Zooming step value to offset zoom from the current position
	Mission Param #4	Focus Locking, Unlocking or Re-locking
	Mission Param #5	Shooting Command
	Mission Param #6	Command Identity
	Mission Param #7	Empty

204	**MAV_CMD_DO_MOUNT_CONFIGURE**	Mission command to configure a camera or antenna mount
	Mission Param #1	Mount operation mode (see MAV_MOUNT_MODE enum)
	Mission Param #2	stabilize roll? (1 = yes, 0 = no)
	Mission Param #3	stabilize pitch? (1 = yes, 0 = no)
	Mission Param #4	stabilize yaw? (1 = yes, 0 = no)
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
205	**MAV_CMD_DO_MOUNT_CONTROL**	Mission command to control a camera or antenna mount
	Mission Param #1	pitch or lat in degrees, depending on mount mode.
	Mission Param #2	roll or lon in degrees depending on mount mode
	Mission Param #3	yaw or alt (in meters) depending on mount mode
	Mission Param #4	reserved
	Mission Param #5	reserved
	Mission Param #6	reserved
	Mission Param #7	MAV_MOUNT_MODE enum value
206	**MAV_CMD_DO_SET_CAM_TRIGG_DIST**	Mission command to set CAM_TRIGG_DIST for this flight
	Mission Param #1	Camera trigger distance (meters)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
207	**MAV_CMD_DO_FENCE_ENABLE**	Mission command to enable the geofence
	Mission Param #1	enable? (0=disable, 1=enable, 2=disable_floor_only)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
208	**MAV_CMD_DO_PARACHUTE**	Mission command to trigger a parachute
	Mission Param #1	action (0=disable, 1=enable, 2=release, for some systems see PARACHUTE_ACTION enum, not in general message set.)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
209	**MAV_CMD_DO_MOTOR_TEST**	Mission command to perform motor test

	Mission Param #1	motor sequence number (a number from 1 to max number of motors on the vehicle)
	Mission Param #2	throttle type (0=throttle percentage, 1=PWM, 2=pilot throttle channel pass-through. See MOTOR_TEST_THROTTLE_TYPE enum)
	Mission Param #3	throttle
	Mission Param #4	timeout (in seconds)
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
210	**MAV_CMD_DO_INVERTED_FLIGHT**	Change to/from inverted flight
	Mission Param #1	inverted (0=normal, 1=inverted)
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
220	**MAV_CMD_DO_MOUNT_CONTROL_QUAT**	Mission command to control a camera or antenna mount, using a quaternion as reference.
	Mission Param #1	q1 - quaternion param #1, w (1 in null-rotation)
	Mission Param #2	q2 - quaternion param #2, x (0 in null-rotation)
	Mission Param #3	q3 - quaternion param #3, y (0 in null-rotation)
	Mission Param #4	q4 - quaternion param #4, z (0 in null-rotation)
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
221	**MAV_CMD_DO_GUIDED_MASTER**	set id of master controller
	Mission Param #1	System ID
	Mission Param #2	Component ID
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
222	**MAV_CMD_DO_GUIDED_LIMITS**	set limits for external control
	Mission Param #1	timeout - maximum time (in seconds) that external controller will be allowed to control vehicle. 0 means no timeout
	Mission Param #2	absolute altitude min (in meters, AMSL) - if vehicle moves below this alt, the command will be aborted and the mission will continue. 0 means no lower altitude limit
	Mission Param #3	absolute altitude max (in meters)- if vehicle moves above this alt, the command will be aborted and the mission will continue. 0 means no upper altitude limit
	Mission Param #4	horizontal move limit (in meters, AMSL) - if vehicle moves more than this distance from it's location at the moment the command was

		executed, the command will be aborted and the mission will continue. 0 means no horizontal altitude limit
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
223	**MAV_CMD_DO_ENGINE_CONTROL**	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
	Mission Param #1	0: Stop engine, 1:Start Engine
	Mission Param #2	0: Warm start, 1:Cold start. Controls use of choke where applicable
	Mission Param #3	Height delay (meters). 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 delay.
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
240	**MAV_CMD_DO_LAST**	NOP - This command is only used to mark the upper limit of the DO commands in the enumeration
	Mission Param #1	Empty
	Mission Param #2	Empty
	Mission Param #3	Empty
	Mission Param #4	Empty
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
241	**MAV_CMD_PREFLIGHT_CALIBRATION**	Trigger calibration. This command will be only accepted if in pre-flight mode.
	Mission Param #1	Gyro calibration: 0: no, 1: yes
	Mission Param #2	Magnetometer calibration: 0: no, 1: yes
	Mission Param #3	Ground pressure: 0: no, 1: yes
	Mission Param #4	Radio calibration: 0: no, 1: yes
	Mission Param #5	Accelerometer calibration: 0: no, 1: yes
	Mission Param #6	Compass/Motor interference calibration: 0: no, 1: yes
	Mission Param #7	Empty
242	**MAV_CMD_PREFLIGHT_SET_SENSOR_OFFSETS**	Set sensor offsets. This command will be only accepted if in pre-flight mode.
	Mission Param #1	Sensor to adjust the offsets for: 0: gyros, 1: accelerometer, 2: magnetometer, 3: barometer, 4: optical flow, 5: second magnetometer, 6: third magnetometer
	Mission Param #2	X axis offset (or generic dimension 1), in the sensor's raw units
	Mission Param #3	Y axis offset (or generic dimension 2), in the sensor's raw units

		sensor's raw units
	Mission Param #4	Z axis offset (or generic dimension 3), in the sensor's raw units
	Mission Param #5	Generic dimension 4, in the sensor's raw units
	Mission Param #6	Generic dimension 5, in the sensor's raw units
	Mission Param #7	Generic dimension 6, in the sensor's raw units
243	**MAV_CMD_PREFLIGHT_UAVCAN**	Trigger UAVCAN config. This command will be only accepted if in pre-flight mode.
	Mission Param #1	1: Trigger actuator ID assignment and direction mapping.
	Mission Param #2	Reserved
	Mission Param #3	Reserved
	Mission Param #4	Reserved
	Mission Param #5	Reserved
	Mission Param #6	Reserved
	Mission Param #7	Reserved
245	**MAV_CMD_PREFLIGHT_STORAGE**	Request storage of different parameter values and logs. This command will be only accepted if in pre-flight mode.
	Mission Param #1	Parameter storage: 0: READ FROM FLASH/EEPROM, 1: WRITE CURRENT TO FLASH/EEPROM, 2: Reset to defaults
	Mission Param #2	Mission storage: 0: READ FROM FLASH/EEPROM, 1: WRITE CURRENT TO FLASH/EEPROM, 2: Reset to defaults
	Mission Param #3	Onboard logging: 0: Ignore, 1: Start default rate logging, -1: Stop logging, > 1: start logging with rate of param 3 in Hz (e.g. set to 1000 for 1000 Hz logging)
	Mission Param #4	Reserved
	Mission Param #5	Empty
	Mission Param #6	Empty
	Mission Param #7	Empty
246	**MAV_CMD_PREFLIGHT_REBOOT_SHUTDOWN**	Request the reboot or shutdown of system components.
	Mission Param #1	0: Do nothing for autopilot, 1: Reboot autopilot, 2: Shutdown autopilot, 3: Reboot autopilot and keep it in the bootloader until upgraded.
	Mission Param #2	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.
	Mission Param #3	Reserved, send 0
	Mission Param #4	Reserved, send 0
	Mission Param #5	Reserved, send 0
	Mission Param #6	Reserved, send 0
	Mission Param #7	Reserved, send 0
252	**MAV_CMD_OVERRIDE_GOTO**	Hold / continue the current action
	Mission Param #1	MAV_GOTO_DO_HOLD: hold MAV_GOTO_DO_CONTINUE: continue with next item in mission plan
	Mission Param #2	MAV_GOTO_HOLD_AT_CURRENT_POSITION: Hold at current position

		Hold at current position MAV_GOTO_HOLD_AT_SPECIFIED_POSITION: hold at specified position
	Mission Param #3	MAV_FRAME coordinate frame of hold point
	Mission Param #4	Desired yaw angle in degrees
	Mission Param #5	Latitude / X position
	Mission Param #6	Longitude / Y position
	Mission Param #7	Altitude / Z position
300	**MAV_CMD_MISSION_START**	start running a mission
	Mission Param #1	first_item: the first mission item to run
	Mission Param #2	last_item: the last mission item to run (after this item is run, the mission ends)
400	**MAV_CMD_COMPONENT_ARM_DISARM**	Arms / Disarms a component
	Mission Param #1	1 to arm, 0 to disarm
410	**MAV_CMD_GET_HOME_POSITION**	Request the home position from the vehicle.
	Mission Param #1	Reserved
	Mission Param #2	Reserved
	Mission Param #3	Reserved
	Mission Param #4	Reserved
	Mission Param #5	Reserved
	Mission Param #6	Reserved
	Mission Param #7	Reserved
500	**MAV_CMD_START_RX_PAIR**	Starts receiver pairing
	Mission Param #1	0:Spektrum
	Mission Param #2	0:Spektrum DSM2, 1:Spektrum DSMX
510	**MAV_CMD_GET_MESSAGE_INTERVAL**	Request the interval between messages for a particular MAVLink message ID
	Mission Param #1	The MAVLink message ID
511	**MAV_CMD_SET_MESSAGE_INTERVAL**	Request the interval between messages for a particular MAVLink message ID. This interface replaces REQUEST_DATA_STREAM
	Mission Param #1	The MAVLink message ID
	Mission Param #2	The interval between two messages, in microseconds. Set to -1 to disable and 0 to request default rate.
520	**MAV_CMD_REQUEST_AUTOPILOT_CAPABILITIES**	Request autopilot capabilities
	Mission Param #1	1: Request autopilot version
	Mission Param #2	Reserved (all remaining params)
2000	**MAV_CMD_IMAGE_START_CAPTURE**	Start image capture sequence
	Mission Param #1	Duration between two consecutive pictures (in seconds)
	Mission Param #2	Number of images to capture total - 0 for unlimited capture
	Mission Param #3	Resolution in megapixels (0.3 for 640x480, 1.3 for 1280x720, etc)

2001	**MAV_CMD_IMAGE_STOP_CAPTURE**	Stop image capture sequence
	Mission Param #1	Reserved
	Mission Param #2	Reserved
2003	**MAV_CMD_DO_TRIGGER_CONTROL**	Enable or disable on-board camera triggering system.
	Mission Param #1	Trigger enable/disable (0 for disable, 1 for start)
	Mission Param #2	Shutter integration time (in ms)
	Mission Param #3	Reserved
2500	**MAV_CMD_VIDEO_START_CAPTURE**	Starts video capture
	Mission Param #1	Camera ID (0 for all cameras), 1 for first, 2 for second, etc.
	Mission Param #2	Frames per second
	Mission Param #3	Resolution in megapixels (0.3 for 640x480, 1.3 for 1280x720, etc)
2501	**MAV_CMD_VIDEO_STOP_CAPTURE**	Stop the current video capture
	Mission Param #1	Reserved
	Mission Param #2	Reserved
2800	**MAV_CMD_PANORAMA_CREATE**	Create a panorama at the current position
	Mission Param #1	Viewing angle horizontal of the panorama (in degrees, +- 0.5 the total angle)
	Mission Param #2	Viewing angle vertical of panorama (in degrees)
	Mission Param #3	Speed of the horizontal rotation (in degrees per second)
	Mission Param #4	Speed of the vertical rotation (in degrees per second)
3000	**MAV_CMD_DO_VTOL_TRANSITION**	Request VTOL transition
	Mission Param #1	The target VTOL state, as defined by ENUM MAV_VTOL_STATE. Only MAV_VTOL_STATE_MC and MAV_VTOL_STATE_FW can be used.
4000	**MAV_CMD_SET_GUIDED_SUBMODE_STANDARD**	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 velocites along all three axes.
4001	**MAV_CMD_SET_GUIDED_SUBMODE_CIRCLE**	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.
	Mission Param #1	Radius of desired circle in CIRCLE_MODE
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Unscaled target latitude of center of circle in CIRCLE_MODE
	Mission Param #6	Unscaled target longitude of center of circle in

		CIRCLE_MODE
30001	**MAV_CMD_PAYLOAD_PREPARE_DEPLOY**	Deploy payload on a Lat / Lon / Alt position. This includes the navigation to reach the required release position and velocity.
	Mission Param #1	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.
	Mission Param #2	Desired approach vector in degrees compass heading (0..360). A negative value indicates the system can define the approach vector at will.
	Mission Param #3	Desired ground speed at release time. This can be overriden by the airframe in case it needs to meet minimum airspeed. A negative value indicates the system can define the ground speed at will.
	Mission Param #4	Minimum altitude clearance to the release position in meters. A negative value indicates the system can define the clearance at will.
	Mission Param #5	Latitude unscaled for MISSION_ITEM or in 1e7 degrees for MISSION_ITEM_INT
	Mission Param #6	Longitude unscaled for MISSION_ITEM or in 1e7 degrees for MISSION_ITEM_INT
	Mission Param #7	Altitude, in meters AMSL
30002	**MAV_CMD_PAYLOAD_CONTROL_DEPLOY**	Control the payload deployment.
	Mission Param #1	Operation mode. 0: Abort deployment, continue normal mission. 1: switch to payload deploment mode. 100: delete first payload deployment request. 101: delete all payload deployment requests.
	Mission Param #2	Reserved
	Mission Param #3	Reserved
	Mission Param #4	Reserved
	Mission Param #5	Reserved
	Mission Param #6	Reserved
	Mission Param #7	Reserved
31000	**MAV_CMD_WAYPOINT_USER_1**	User defined waypoint item. Ground Station will show the Vehicle as flying through this item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled
	Mission Param #7	Altitude, in meters AMSL
31001	**MAV_CMD_WAYPOINT_USER_2**	User defined waypoint item. Ground Station will show the Vehicle as flying through this item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined

	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled
	Mission Param #7	Altitude, in meters AMSL
31002	**MAV_CMD_WAYPOINT_USER_3**	User defined waypoint item. Ground Station will show the Vehicle as flying through this item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled
	Mission Param #7	Altitude, in meters AMSL
31003	**MAV_CMD_WAYPOINT_USER_4**	User defined waypoint item. Ground Station will show the Vehicle as flying through this item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled
	Mission Param #7	Altitude, in meters AMSL
31004	**MAV_CMD_WAYPOINT_USER_5**	User defined waypoint item. Ground Station will show the Vehicle as flying through this item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled
	Mission Param #7	Altitude, in meters AMSL
31005	**MAV_CMD_SPATIAL_USER_1**	User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled
	Mission Param #7	Altitude, in meters AMSL
31006	**MAV_CMD_SPATIAL_USER_2**	User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled

	Mission Param #7	Altitude, in meters AMSL
31007	**MAV_CMD_SPATIAL_USER_3**	User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled
	Mission Param #7	Altitude, in meters AMSL
31008	**MAV_CMD_SPATIAL_USER_4**	User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled
	Mission Param #7	Altitude, in meters AMSL
31009	**MAV_CMD_SPATIAL_USER_5**	User defined spatial item. Ground Station will not show the Vehicle as flying through this item. Example: ROI item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	Latitude unscaled
	Mission Param #6	Longitude unscaled
	Mission Param #7	Altitude, in meters AMSL
31010	**MAV_CMD_USER_1**	User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	User defined
	Mission Param #6	User defined
	Mission Param #7	User defined
31011	**MAV_CMD_USER_2**	User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined

	Mission Param #5	User defined
	Mission Param #6	User defined
	Mission Param #7	User defined
31012	**MAV_CMD_USER_3**	User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	User defined
	Mission Param #6	User defined
	Mission Param #7	User defined
31013	**MAV_CMD_USER_4**	User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	User defined
	Mission Param #6	User defined
	Mission Param #7	User defined
31014	**MAV_CMD_USER_5**	User defined command. Ground Station will not show the Vehicle as flying through this item. Example: MAV_CMD_DO_SET_PARAMETER item.
	Mission Param #1	User defined
	Mission Param #2	User defined
	Mission Param #3	User defined
	Mission Param #4	User defined
	Mission Param #5	User defined
	Mission Param #6	User defined
	Mission Param #7	User defined

MAV_DATA_STREAM

THIS INTERFACE IS DEPRECATED AS OF JULY 2015. Please use MESSAGE_INTERVAL instead. 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.

CMD ID	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/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

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).

CMD ID	Field Name	Description
0	**MAV_ROI_NONE**	No region of interest.
1	**MAV_ROI_WPNEXT**	Point toward next MISSION.
2	**MAV_ROI_WPINDEX**	Point toward given MISSION.
3	**MAV_ROI_LOCATION**	Point toward fixed location.
4	**MAV_ROI_TARGET**	Point toward of given id.

MAV_CMD_ACK

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

CMD ID	Field Name	Description
	MAV_CMD_ACK_OK	Command / mission item is ok.

MAV_CMD_ACK_ERR_FAIL	Generic error message if none of the other reasons fails or if no detailed error reporting is implemented.
MAV_CMD_ACK_ERR_ACCESS_DENIED	The system is refusing to accept this command from this source / communication partner.
MAV_CMD_ACK_ERR_NOT_SUPPORTED	Command or mission item is not supported, other commands would be accepted.
MAV_CMD_ACK_ERR_COORDINATE_FRAME_NOT_SUPPORTED	The coordinate frame of this command / mission item is not supported.
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.
MAV_CMD_ACK_ERR_X_LAT_OUT_OF_RANGE	The X or latitude value is out of range.
MAV_CMD_ACK_ERR_Y_LON_OUT_OF_RANGE	The Y or longitude value is out of range.
MAV_CMD_ACK_ERR_Z_ALT_OUT_OF_RANGE	The Z or altitude value is out of range.

MAV_PARAM_TYPE

Specifies the datatype of a MAVLink parameter.

CMD ID	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_RESULT

result from a mavlink command

CMD ID	Field Name	Description
0	**MAV_RESULT_ACCEPTED**	Command ACCEPTED and EXECUTED
1	**MAV_RESULT_TEMPORARILY_REJECTED**	Command TEMPORARY REJECTED/DENIED
2	**MAV_RESULT_DENIED**	Command PERMANENTLY DENIED
3	**MAV_RESULT_UNSUPPORTED**	Command UNKNOWN/UNSUPPORTED
4	**MAV_RESULT_FAILED**	Command executed, but failed

MAV_MISSION_RESULT

result in a mavlink mission ack

CMD ID	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 item exceeds 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**	param7 has an invalid value
13	**MAV_MISSION_INVALID_SEQUENCE**	received waypoint out of sequence
14	**MAV_MISSION_DENIED**	not accepting any mission commands from this communication partner

MAV_SEVERITY

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/.

CMD ID	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 occured, 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

Power supply status flags (bitmask)

CMD ID	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

SERIAL_CONTROL device types

CMD ID	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

SERIAL_CONTROL_FLAG

SERIAL_CONTROL flags (bitmask)

CMD ID	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

Enumeration of distance sensor types

CMD ID	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

MAV_SENSOR_ORIENTATION

Enumeration of sensor orientation, according to its rotations

CMD ID	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_315_PITCH_315_YAW_315**	Roll: 315, Pitch: 315, Yaw: 315

MAV_PROTOCOL_CAPABILITY

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

CMD ID	Field Name	Description
1	**MAV_PROTOCOL_CAPABILITY_MISSION_FLOAT**	Autopilot supports MISSION float message type.
2	**MAV_PROTOCOL_CAPABILITY_PARAM_FLOAT**	Autopilot supports the new param float message type.
4	**MAV_PROTOCOL_CAPABILITY_MISSION_INT**	Autopilot supports MISSION_INT scaled integer message type.
8	**MAV_PROTOCOL_CAPABILITY_COMMAND_INT**	Autopilot supports COMMAND_INT scaled integer message type.
16	**MAV_PROTOCOL_CAPABILITY_PARAM_UNION**	Autopilot supports the new param union message type.
32	**MAV_PROTOCOL_CAPABILITY_FTP**	Autopilot supports the new FILE_TRANSFER_PROTOCOL message type.
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 flight termination command.
4096	**MAV_PROTOCOL_CAPABILITY_COMPASS_CALIBRATION**	Autopilot supports onboard compass calibration.

MAV_ESTIMATOR_TYPE

Enumeration of estimator types

CMD ID	Field Name	Description
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.

MAV_BATTERY_TYPE

Enumeration of battery types

CMD ID	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

Enumeration of battery functions

CMD ID	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_VTOL_STATE

Enumeration of VTOL states

CMD ID	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

Enumeration of landed detector states

CMD ID	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

ADSB_ALTITUDE_TYPE

Enumeration of the ADSB altimeter types

CMD ID	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

ADSB classification for the type of vehicle emitting the transponder signal

CMD ID	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

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

CMD ID	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**	

MAV_DO_REPOSITION_FLAGS

Bitmask of options for the MAV_CMD_DO_REPOSITION

CMD ID	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

Flags in EKF_STATUS message

CMD ID	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

MOTOR_TEST_THROTTLE_TYPE

CMD ID	Field Name	Description
0	**MOTOR_TEST_THROTTLE_PERCENT**	throttle as a percentage from 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

GPS_INPUT_IGNORE_FLAGS

CMD ID	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

Possible actions an aircraft can take to avoid a collision.

CMD ID	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 thread
3	MAV_COLLISION_ACTION_MOVE_HORIZONTALLY	Ascend or Descend to avoid thread
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

Aircraft-rated danger from this threat.

CMD ID	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 panicing, and may take actions to avoid threat

MAV_COLLISION_SRC

Source of information about this collision.

CMD ID	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

MAVLink Messages

HEARTBEAT (#0)

The heartbeat message shows that a system is present and responding. The type of the MAV and Autopilot hardware allow the receiving system to treat further messages from this system appropriate (e.g. by laying out the user interface based on the autopilot).

Field Name	Type	Description
type	uint8_t	Type of the MAV (quadrotor, helicopter, etc., up to 15 types, defined in MAV_TYPE ENUM)
autopilot	uint8_t	Autopilot type / class. defined in MAV_AUTOPILOT ENUM
base_mode	uint8_t	System mode bitfield, see MAV_MODE_FLAG ENUM in mavlink/include/mavlink_types.h
custom_mode	uint32_t	A bitfield for use for autopilot-specific flags.
system_status	uint8_t	System status flag, see MAV_STATE ENUM
mavlink_version	uint8_t_mavlink_version	MAVLink version, not writable by user, gets added by protocol because of magic data type: uint8_t_mavlink_version

SYS_STATUS (#1)

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 wether the system is currently active or not and if an emergency occured. During the CRITICAL and EMERGENCY states the MAV is still considered to be active, but should start emergency procedures autonomously. After a failure occured it should first move from active to critical to allow manual intervention and then move to emergency after a certain timeout.

Field Name	Type	Description
onboard_control_sensors_present	uint32_t	Bitmask showing which onboard controllers and sensors are present. Value of 0: not present. Value of 1: present. Indices defined by ENUM MAV_SYS_STATUS_SENSOR
onboard_control_sensors_enabled	uint32_t	Bitmask showing which onboard controllers and sensors are enabled: Value of 0: not enabled. Value of 1: enabled. Indices defined by ENUM MAV_SYS_STATUS_SENSOR
onboard_control_sensors_health	uint32_t	Bitmask showing which onboard controllers and sensors are operational or have an error: Value of 0: not enabled. Value of 1: enabled. Indices defined by ENUM MAV_SYS_STATUS_SENSOR
load	uint16_t	Maximum usage in percent of the mainloop time, (0%: 0, 100%: 1000) should be always below 1000
voltage_battery	uint16_t	Battery voltage, in millivolts (1 = 1 millivolt)
current_battery	int16_t	Battery current, in 10*milliamperes (1 = 10 milliampere), -1: autopilot does not measure the current
battery_remaining	int8_t	Remaining battery energy: (0%: 0, 100%: 100), -1: autopilot estimate the remaining battery
drop_rate_comm	uint16_t	Communication drops in percent, (0%: 0, 100%: 10'000), (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

SYSTEM_TIME (#2)

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

Field Name	Type	Description
time_unix_usec	uint64_t	Timestamp of the master clock in microseconds since UNIX epoch.
time_boot_ms	uint32_t	Timestamp of the component clock since boot time in milliseconds.

PING (#4)

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.

Field Name	Type	Description
time_usec	uint64_t	Unix timestamp in microseconds or since system boot if smaller than MAVLink epoch (1.1.2009)
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 system id of the requesting system

CHANGE_OPERATOR_CONTROL (#5)

Request to control this MAV

Field Name	Type	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	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)

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)

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

SET_MODE (#11)

THIS INTERFACE IS DEPRECATED. USE COMMAND_LONG with MAV_CMD_DO_SET_MODE INSTEAD. 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	Description
target_system	uint8_t	The system setting the mode
base_mode	uint8_t	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)

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 http://qgroundcontrol.org/parameter_interface 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)

Request all parameters of this component. After this request, all parameters are emitted.

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

PARAM_VALUE (#22)

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.

Field Name	Type	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	Onboard parameter type: see the MAV_PARAM_TYPE enum for supported data types.
param_count	uint16_t	Total number of onboard parameters
param_index	uint16_t	Index of this onboard parameter

PARAM_SET (#23)

Set a parameter value TEMPORARILY to RAM. It will be reset to default on system reboot. Send the ACTION MAV_ACTION_STORAGE_WRITE to PERMANENTLY write the RAM contents to EEPROM. IMPORTANT: The receiving component should acknowledge the new parameter value by sending a param_value message to all communication partners. This will also ensure that multiple GCS all have an up-to-date list of all parameters. If the sending GCS did not receive a PARAM_VALUE message within its timeout time, it should re-send the PARAM_SET message.

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_value	float	Onboard parameter value
param_type	uint8_t	Onboard parameter type: see the MAV_PARAM_TYPE enum for supported data types.

GPS_RAW_INT (#24)

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 for the global position estimate. Coordinate frame is right-handed, Z-axis up (GPS frame).

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds since UNIX epoch or microseconds since system boot)
fix_type	uint8_t	0-1: no fix, 2: 2D fix, 3: 3D fix, 4: DGPS, 5: RTK. 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	Latitude (WGS84), in degrees * 1E7
lon	int32_t	Longitude (WGS84), in degrees * 1E7
alt	int32_t	Altitude (AMSL, NOT WGS84), in meters * 1000 (positive for up). Note that virtually all GPS modules provide the AMSL altitude in addition to the WGS84 altitude.
eph	uint16_t	GPS HDOP horizontal dilution of position (unitless). If unknown, set to: UINT16_MAX
epv	uint16_t	GPS VDOP vertical dilution of position (unitless). If unknown, set to: UINT16_MAX
vel	uint16_t	GPS ground speed (m/s * 100). If unknown, set to: UINT16_MAX
cog	uint16_t	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 255

GPS_STATUS (#25)

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 for the global position estimate. This message can contain information for up to 20 satellites.

Field Name	Type	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]	Elevation (0: right on top of receiver, 90: on the horizon) of satellite
satellite_azimuth	uint8_t[20]	Direction of satellite, 0: 0 deg, 255: 360 deg.
satellite_snr	uint8_t[20]	Signal to noise ratio of satellite

SCALED_IMU (#26)

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

Field Name	Type	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
xacc	int16_t	X acceleration (mg)
yacc	int16_t	Y acceleration (mg)
zacc	int16_t	Z acceleration (mg)
xgyro	int16_t	Angular speed around X axis (millirad /sec)
ygyro	int16_t	Angular speed around Y axis (millirad /sec)
zgyro	int16_t	Angular speed around Z axis (millirad /sec)
xmag	int16_t	X Magnetic field (milli tesla)
ymag	int16_t	Y Magnetic field (milli tesla)
zmag	int16_t	Z Magnetic field (milli tesla)

RAW_IMU (#27)

The RAW IMU readings for the usual 9DOF sensor setup. This message should always contain the true raw values without any scaling to allow data capture and system debugging.

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds since UNIX epoch or microseconds since system boot)
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)

RAW_PRESSURE (#28)

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	Description
time_usec	uint64_t	Timestamp (microseconds since UNIX epoch or microseconds since system boot)
press_abs	int16_t	Absolute pressure (raw)
press_diff1	int16_t	Differential pressure 1 (raw, 0 if nonexistant)
press_diff2	int16_t	Differential pressure 2 (raw, 0 if nonexistant)
temperature	int16_t	Raw Temperature measurement (raw)

SCALED_PRESSURE (#29)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
press_abs	float	Absolute pressure (hectopascal)
press_diff	float	Differential pressure 1 (hectopascal)
temperature	int16_t	Temperature measurement (0.01 degrees celsius)

ATTITUDE (#30)

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

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

ATTITUDE_QUATERNION (#31)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds 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	Roll angular speed (rad/s)
pitchspeed	float	Pitch angular speed (rad/s)
yawspeed	float	Yaw angular speed (rad/s)

LOCAL_POSITION_NED (#32)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
x	float	X Position
y	float	Y Position
z	float	Z Position
vx	float	X Speed
vy	float	Y Speed
vz	float	Z Speed

GLOBAL_POSITION_INT (#33)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
lat	int32_t	Latitude, expressed as degrees * 1E7
lon	int32_t	Longitude, expressed as degrees * 1E7
alt	int32_t	Altitude in meters, expressed as * 1000 (millimeters), AMSL (not WGS84 - note that virtually all GPS modules provide the AMSL as well)
relative_alt	int32_t	Altitude above ground in meters, expressed as * 1000 (millimeters)
vx	int16_t	Ground X Speed (Latitude, positive north), expressed as m/s * 100
vy	int16_t	Ground Y Speed (Longitude, positive east), expressed as m/s * 100
vz	int16_t	Ground Z Speed (Altitude, positive down), expressed as m/s * 100
hdg	uint16_t	Vehicle heading (yaw angle) in degrees * 100, 0.0..359.99 degrees. If unknown, set to: UINT16_MAX

RC_CHANNELS_SCALED (#34)

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	Description

time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
port	uint8_t	Servo output port (set of 8 outputs = 1 port). Most MAVs will just use one, but this allows for more than 8 servos.
chan1_scaled	int16_t	RC channel 1 value scaled, (-100%) -10000, (0%) 0, (100%) 10000, (invalid) INT16_MAX.
chan2_scaled	int16_t	RC channel 2 value scaled, (-100%) -10000, (0%) 0, (100%) 10000, (invalid) INT16_MAX.
chan3_scaled	int16_t	RC channel 3 value scaled, (-100%) -10000, (0%) 0, (100%) 10000, (invalid) INT16_MAX.
chan4_scaled	int16_t	RC channel 4 value scaled, (-100%) -10000, (0%) 0, (100%) 10000, (invalid) INT16_MAX.
chan5_scaled	int16_t	RC channel 5 value scaled, (-100%) -10000, (0%) 0, (100%) 10000, (invalid) INT16_MAX.
chan6_scaled	int16_t	RC channel 6 value scaled, (-100%) -10000, (0%) 0, (100%) 10000, (invalid) INT16_MAX.
chan7_scaled	int16_t	RC channel 7 value scaled, (-100%) -10000, (0%) 0, (100%) 10000, (invalid) INT16_MAX.
chan8_scaled	int16_t	RC channel 8 value scaled, (-100%) -10000, (0%) 0, (100%) 10000, (invalid) INT16_MAX.
rssi	uint8_t	Receive signal strength indicator, 0: 0%, 100: 100%, 255: invalid/unknown.

RC_CHANNELS_RAW (#35)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
port	uint8_t	Servo output port (set of 8 outputs = 1 port). Most MAVs will just use one, but this allows for more than 8 servos.
chan1_raw	uint16_t	RC channel 1 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan2_raw	uint16_t	RC channel 2 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan3_raw	uint16_t	RC channel 3 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan4_raw	uint16_t	RC channel 4 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan5_raw	uint16_t	RC channel 5 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan6_raw	uint16_t	RC channel 6 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan7_raw	uint16_t	RC channel 7 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan8_raw	uint16_t	RC channel 8 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
rssi	uint8_t	Receive signal strength indicator, 0: 0%, 100: 100%, 255: invalid/unknown.

SERVO_OUTPUT_RAW (#36)

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	Description
time_usec	uint32_t	Timestamp (microseconds since system boot)
port	uint8_t	Servo output port (set of 8 outputs = 1 port). Most MAVs will just use one, but this allows to encode more than 8 servos.
servo1_raw	uint16_t	Servo output 1 value, in microseconds
servo2_raw	uint16_t	Servo output 2 value, in microseconds
servo3_raw	uint16_t	Servo output 3 value, in microseconds
servo4_raw	uint16_t	Servo output 4 value, in microseconds
servo5_raw	uint16_t	Servo output 5 value, in microseconds
servo6_raw	uint16_t	Servo output 6 value, in microseconds
servo7_raw	uint16_t	Servo output 7 value, in microseconds
servo8_raw	uint16_t	Servo output 8 value, in microseconds
servo9_raw	uint16_t	Servo output 9 value, in microseconds
servo10_raw	uint16_t	Servo output 10 value, in microseconds
servo11_raw	uint16_t	Servo output 11 value, in microseconds
servo12_raw	uint16_t	Servo output 12 value, in microseconds
servo13_raw	uint16_t	Servo output 13 value, in microseconds
servo14_raw	uint16_t	Servo output 14 value, in microseconds
servo15_raw	uint16_t	Servo output 15 value, in microseconds
servo16_raw	uint16_t	Servo output 16 value, in microseconds

MISSION_REQUEST_PARTIAL_LIST (#37)

Request a partial list of mission items from the system/component. http://qgroundcontrol.org/mavlink/waypoint_protocol. If start and end index are the same, just send one waypoint.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
start_index	int16_t	Start index, 0 by default
end_index	int16_t	End index, -1 by default (-1: send list to end). Else a valid index of the list

MISSION_WRITE_PARTIAL_LIST (#38)

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	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
start_index	int16_t	Start index, 0 by default and smaller / equal to the largest index of the current onboard list.
end_index	int16_t	End index, equal or greater than start index.

MISSION_ITEM (#39)

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). See also http://qgroundcontrol.org/mavlink/waypoint_protocol.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
seq	uint16_t	Sequence
frame	uint8_t	The coordinate system of the MISSION. see MAV_FRAME in mavlink_types.h
command	uint16_t	The scheduled action for the MISSION. see MAV_CMD in common.xml MAVLink specs
current	uint8_t	false:0, true:1
autocontinue	uint8_t	autocontinue to next wp
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 position, global: latitude
y	float	PARAM6 / y position: global: longitude
z	float	PARAM7 / z position: global: altitude (relative or absolute, depending on frame.

MISSION_REQUEST (#40)

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. http://qgroundcontrol.org/mavlink/waypoint_protocol

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

MISSION_SET_CURRENT (#41)

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 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)

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

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

MISSION_COUNT (#44)

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 MISSIONs.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
count	uint16_t	Number of mission items in the sequence

MISSION_CLEAR_ALL (#45)

Delete all mission items at once.

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

MISSION_ITEM_REACHED (#46)

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 MISSION.

Field Name	Type	Description
seq	uint16_t	Sequence

MISSION_ACK (#47)

Ack message during MISSION 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	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
type	uint8_t	See MAV_MISSION_RESULT enum

SET_GPS_GLOBAL_ORIGIN (#48)

As local waypoints exist, the global MISSION reference allows to transform between the local coordinate frame and the global (GPS) coordinate frame. This can be necessary when e.g. in- and outdoor settings are connected and the MAV should move from in- to outdoor.

Field Name	Type	Description
target_system	uint8_t	System ID
latitude	int32_t	Latitude (WGS84), in degrees * 1E7
longitude	int32_t	Longitude (WGS84, in degrees * 1E7
altitude	int32_t	Altitude (AMSL), in meters * 1000 (positive for up)

GPS_GLOBAL_ORIGIN (#49)

Once the MAV sets a new GPS-Local correspondence, this message announces the origin (0,0,0) position

Field Name	Type	Description
latitude	int32_t	Latitude (WGS84), in degrees * 1E7
longitude	int32_t	Longitude (WGS84), in degrees * 1E7
altitude	int32_t	Altitude (AMSL), in meters * 1000 (positive for up)

PARAM_MAP_RC (#50)

Bind a RC channel to a parameter. The parameter should change accoding 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 correpsonds 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)

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.
http://qgroundcontrol.org/mavlink/waypoint_protocol

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

SAFETY_SET_ALLOWED_AREA (#54)

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/MISSIONs to accept and which to reject. Safety areas are often enforced by national or competition regulations.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
frame	uint8_t	Coordinate frame, as defined by MAV_FRAME enum in mavlink_types.h. Can be either global, GPS, right-handed with Z axis up or local, right handed, Z axis down.
p1x	float	x position 1 / Latitude 1
p1y	float	y position 1 / Longitude 1
p1z	float	z position 1 / Altitude 1
p2x	float	x position 2 / Latitude 2
p2y	float	y position 2 / Longitude 2
p2z	float	z position 2 / Altitude 2

SAFETY_ALLOWED_AREA (#55)

Read out the safety zone the MAV currently assumes.

Field Name	Type	Description
frame	uint8_t	Coordinate frame, as defined by MAV_FRAME enum in mavlink_types.h. Can be either global, GPS, right-handed with Z axis up or local, right handed, Z axis down.

p1x	float	x position 1 / Latitude 1
p1y	float	y position 1 / Longitude 1
p1z	float	z position 1 / Altitude 1
p2x	float	x position 2 / Latitude 2
p2y	float	y position 2 / Longitude 2
p2z	float	z position 2 / Altitude 2

ATTITUDE_QUATERNION_COV (#61)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
q	float[4]	Quaternion components, w, x, y, z (1 0 0 0 is the null-rotation)
rollspeed	float	Roll angular speed (rad/s)
pitchspeed	float	Pitch angular speed (rad/s)
yawspeed	float	Yaw angular speed (rad/s)
covariance	float[9]	Attitude covariance

NAV_CONTROLLER_OUTPUT (#62)

The state of the fixed wing navigation and position controller.

Field Name	Type	Description
nav_roll	float	Current desired roll in degrees
nav_pitch	float	Current desired pitch in degrees
nav_bearing	int16_t	Current desired heading in degrees
target_bearing	int16_t	Bearing to current MISSION/target in degrees
wp_dist	uint16_t	Distance to active MISSION in meters
alt_error	float	Current altitude error in meters
aspd_error	float	Current airspeed error in meters/second
xtrack_error	float	Current crosstrack error on x-y plane in meters

GLOBAL_POSITION_INT_COV (#63)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
time_utc	uint64_t	Timestamp (microseconds since UNIX epoch) in UTC. 0 for unknown. Commonly filled by the precision time source of a GPS receiver.

estimator_type	uint8_t	Class id of the estimator this estimate originated from.
lat	int32_t	Latitude, expressed as degrees * 1E7
lon	int32_t	Longitude, expressed as degrees * 1E7
alt	int32_t	Altitude in meters, expressed as * 1000 (millimeters), above MSL
relative_alt	int32_t	Altitude above ground in meters, expressed as * 1000 (millimeters)
vx	float	Ground X Speed (Latitude), expressed as m/s
vy	float	Ground Y Speed (Longitude), expressed as m/s
vz	float	Ground Z Speed (Altitude), expressed as m/s
covariance	float[36]	Covariance matrix (first six entries are the first ROW, next six entries are the second row, etc.)

LOCAL_POSITION_NED_COV (#64)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot). 0 for system without monotonic timestamp
time_utc	uint64_t	Timestamp (microseconds since UNIX epoch) in UTC. 0 for unknown. Commonly filled by the precision time source of a GPS receiver.
estimator_type	uint8_t	Class id of the estimator this estimate originated from.
x	float	X Position
y	float	Y Position
z	float	Z Position
vx	float	X Speed (m/s)
vy	float	Y Speed (m/s)
vz	float	Z Speed (m/s)
ax	float	X Acceleration (m/s^2)
ay	float	Y Acceleration (m/s^2)
az	float	Z Acceleration (m/s^2)
covariance	float[45]	Covariance matrix upper right triangular (first nine entries are the first ROW, next eight entries are the second row, etc.)

RC_CHANNELS (#65)

The PPM 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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds 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	RC channel 1 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan2_raw	uint16_t	RC channel 2 value, in microseconds. A value of UINT16_MAX

		implies the channel is unused.
chan3_raw	uint16_t	RC channel 3 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan4_raw	uint16_t	RC channel 4 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan5_raw	uint16_t	RC channel 5 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan6_raw	uint16_t	RC channel 6 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan7_raw	uint16_t	RC channel 7 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan8_raw	uint16_t	RC channel 8 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan9_raw	uint16_t	RC channel 9 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan10_raw	uint16_t	RC channel 10 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan11_raw	uint16_t	RC channel 11 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan12_raw	uint16_t	RC channel 12 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan13_raw	uint16_t	RC channel 13 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan14_raw	uint16_t	RC channel 14 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan15_raw	uint16_t	RC channel 15 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan16_raw	uint16_t	RC channel 16 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan17_raw	uint16_t	RC channel 17 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
chan18_raw	uint16_t	RC channel 18 value, in microseconds. A value of UINT16_MAX implies the channel is unused.
rssi	uint8_t	Receive signal strength indicator, 0: 0%, 100: 100%, 255: invalid/unknown.

REQUEST_DATA_STREAM (#66)

THIS INTERFACE IS DEPRECATED. USE SET_MESSAGE_INTERVAL INSTEAD.

Field Name	Type	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	The requested message rate
start_stop	uint8_t	1 to start sending, 0 to stop sending.

DATA_STREAM (#67)

THIS INTERFACE IS DEPRECATED. USE MESSAGE_INTERVAL INSTEAD.

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

MANUAL_CONTROL (#69)

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 an buttons are also transmit as boolean values of their

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' current state, 1 for pressed, 0 for released. The lowest bit corresponds to Button 1.

RC_CHANNELS_OVERRIDE (#70)

The RAW values of the RC channels sent to the MAV to override info received from the RC radio. A value of UINT16_MAX means no change to that channel. A value of 0 means control of that channel should be released back to the RC radio. The standard PPM modulation is as follows: 1000 microseconds: 0%, 2000 microseconds: 100%. Individual receivers/transmitters might violate this specification.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
chan1_raw	uint16_t	RC channel 1 value, in microseconds. A value of UINT16_MAX means to ignore this field.
chan2_raw	uint16_t	RC channel 2 value, in microseconds. A value of UINT16_MAX means to ignore this field.
chan3_raw	uint16_t	RC channel 3 value, in microseconds. A value of UINT16_MAX means to ignore this field.
chan4_raw	uint16_t	RC channel 4 value, in microseconds. A value of UINT16_MAX means to ignore this field.
chan5_raw	uint16_t	RC channel 5 value, in microseconds. A value of UINT16_MAX

		means to ignore this field.
chan6_raw	uint16_t	RC channel 6 value, in microseconds. A value of UINT16_MAX means to ignore this field.
chan7_raw	uint16_t	RC channel 7 value, in microseconds. A value of UINT16_MAX means to ignore this field.
chan8_raw	uint16_t	RC channel 8 value, in microseconds. A value of UINT16_MAX means to ignore this field.

MISSION_ITEM_INT (#73)

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). See alsohttp://qgroundcontrol.org/mavlink/waypoint_protocol.

Field Name	Type	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	The coordinate system of the MISSION. see MAV_FRAME in mavlink_types.h
command	uint16_t	The scheduled action for the MISSION. see MAV_CMD in common.xml MAVLink specs
current	uint8_t	false:0, true:1
autocontinue	uint8_t	autocontinue to next wp
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.

VFR_HUD (#74)

Metrics typically displayed on a HUD for fixed wing aircraft

Field Name	Type	Description
airspeed	float	Current airspeed in m/s
groundspeed	float	Current ground speed in m/s
heading	int16_t	Current heading in degrees, in compass units (0..360, 0=north)
throttle	uint16_t	Current throttle setting in integer percent, 0 to 100
alt	float	Current altitude (MSL), in meters
climb	float	Current climb rate in meters/second

COMMAND_INT (#75)

Message encoding a command with parameters as scaled integers. Scaling depends on the actual command value.

Field Name	Type	Description
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
frame	uint8_t	The coordinate system of the COMMAND. see MAV_FRAME in mavlink_types.h
command	uint16_t	The scheduled action for the mission item. see MAV_CMD in common.xml MAVLink specs
current	uint8_t	false:0, true:1
autocontinue	uint8_t	autocontinue to next wp
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)

Send a command with up to seven parameters to the MAV

Field Name	Type	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	Command ID, as defined by MAV_CMD enum.
confirmation	uint8_t	0: First transmission of this command. 1-255: Confirmation transmissions (e.g. for kill command)
param1	float	Parameter 1, as defined by MAV_CMD enum.
param2	float	Parameter 2, as defined by MAV_CMD enum.
param3	float	Parameter 3, as defined by MAV_CMD enum.
param4	float	Parameter 4, as defined by MAV_CMD enum.
param5	float	Parameter 5, as defined by MAV_CMD enum.
param6	float	Parameter 6, as defined by MAV_CMD enum.
param7	float	Parameter 7, as defined by MAV_CMD enum.

COMMAND_ACK (#77)

Report status of a command. Includes feedback wether the command was executed.

Field Name	Type	Description
command	uint16_t	Command ID, as defined by MAV_CMD enum.

result	uint8_t	See MAV_RESULT enum

MANUAL_SETPOINT (#81)

Setpoint in roll, pitch, yaw and thrust from the operator

Field Name	Type	Description
time_boot_ms	uint32_t	Timestamp in milliseconds since system boot
roll	float	Desired roll rate in radians per second
pitch	float	Desired pitch rate in radians per second
yaw	float	Desired yaw rate in radians per second
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)

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

Field Name	Type	Description
time_boot_ms	uint32_t	Timestamp in milliseconds since system boot
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
type_mask	uint8_t	Mappings: If any of these bits are set, the corresponding input should be ignored: bit 1: body roll rate, bit 2: body pitch rate, bit 3: body yaw rate. bit 4-bit 6: reserved, bit 7: throttle, bit 8: attitude
q	float[4]	Attitude quaternion (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
body_roll_rate	float	Body roll rate in radians per second
body_pitch_rate	float	Body roll rate in radians per second
body_yaw_rate	float	Body roll rate in radians per second
thrust	float	Collective thrust, normalized to 0 .. 1 (-1 .. 1 for vehicles capable of reverse trust)

ATTITUDE_TARGET (#83)

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	Description
time_boot_ms	uint32_t	Timestamp in milliseconds since system boot
type_mask	uint8_t	Mappings: If any of these bits are set, the corresponding input should be ignored: bit 1: body roll rate, bit 2: body pitch rate, bit 3: body yaw rate. bit 4-bit 7: reserved, bit 8: attitude
q	float[4]	Attitude quaternion (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
body_roll_rate	float	Body roll rate in radians per second
body_pitch_rate	float	Body roll rate in radians per second

body_pitch_rate	float	Body roll rate in radians per second
body_yaw_rate	float	Body roll rate in radians per second
thrust	float	Collective thrust, normalized to 0 .. 1 (-1 .. 1 for vehicles capable of reverse trust)

SET_POSITION_TARGET_LOCAL_NED (#84)

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	Description
time_boot_ms	uint32_t	Timestamp in milliseconds since system boot
target_system	uint8_t	System ID
target_component	uint8_t	Component ID
coordinate_frame	uint8_t	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	Bitmask 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 10 is set the floats afx afy afz should be interpreted as force instead of acceleration. Mapping: bit 1: x, bit 2: y, bit 3: z, bit 4: vx, bit 5: vy, bit 6: vz, bit 7: ax, bit 8: ay, bit 9: az, bit 10: is force setpoint, bit 11: yaw, bit 12: yaw rate
x	float	X Position in NED frame in meters
y	float	Y Position in NED frame in meters
z	float	Z Position in NED frame in meters (note, altitude is negative in NED)
vx	float	X velocity in NED frame in meter / s
vy	float	Y velocity in NED frame in meter / s
vz	float	Z velocity in NED frame in meter / s
afx	float	X acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afy	float	Y acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afz	float	Z acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
yaw	float	yaw setpoint in rad
yaw_rate	float	yaw rate setpoint in rad/s

POSITION_TARGET_LOCAL_NED (#85)

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	Description
time_boot_ms	uint32_t	Timestamp in milliseconds since system boot
coordinate_frame	uint8_t	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	Bitmask 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 10 is set the floats afx afy afz should be interpreted as force instead of acceleration. Mapping: bit 1: x, bit 2: y, bit 3: z, bit 4: vx, bit 5: vy, bit 6: vz, bit 7: ax, bit 8: ay, bit 9: az, bit 10: is force setpoint, bit 11: yaw, bit 12: yaw rate
x	float	X Position in NED frame in meters
y	float	Y Position in NED frame in meters
z	float	Z Position in NED frame in meters (note, altitude is negative in NED)
vx	float	X velocity in NED frame in meter / s
vy	float	Y velocity in NED frame in meter / s
vz	float	Z velocity in NED frame in meter / s
afx	float	X acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afy	float	Y acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afz	float	Z acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
yaw	float	yaw setpoint in rad
yaw_rate	float	yaw rate setpoint in rad/s

SET_POSITION_TARGET_GLOBAL_INT (#86)

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	Description
time_boot_ms	uint32_t	Timestamp in milliseconds 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	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	Bitmask 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 10 is set the floats afx afy afz should be interpreted as force instead of acceleration. Mapping: bit 1: x, bit 2: y, bit 3: z, bit 4: vx, bit 5: vy, bit 6: vz, bit 7: ax, bit 8: ay, bit 9: az, bit 10: is force setpoint, bit 11: yaw, bit 12: yaw rate
lat_int	int32_t	X Position in WGS84 frame in 1e7 * meters
lon_int	int32_t	Y Position in WGS84 frame in 1e7 * meters
alt	float	Altitude in meters in AMSL altitude, not WGS84 if absolute or relative, above terrain if GLOBAL_TERRAIN_ALT_INT
vx	float	X velocity in NED frame in meter / s
vy	float	Y velocity in NED frame in meter / s
vz	float	Z velocity in NED frame in meter / s
afx	float	X acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N

afy	float	Y acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afz	float	Z acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
yaw	float	yaw setpoint in rad
yaw_rate	float	yaw rate setpoint in rad/s

POSITION_TARGET_GLOBAL_INT (#87)

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	Description
time_boot_ms	uint32_t	Timestamp in milliseconds 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	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	Bitmask 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 10 is set the floats afx afy afz should be interpreted as force instead of acceleration. Mapping: bit 1: x, bit 2: y, bit 3: z, bit 4: vx, bit 5: vy, bit 6: vz, bit 7: ax, bit 8: ay, bit 9: az, bit 10: is force setpoint, bit 11: yaw, bit 12: yaw rate
lat_int	int32_t	X Position in WGS84 frame in 1e7 * meters
lon_int	int32_t	Y Position in WGS84 frame in 1e7 * meters
alt	float	Altitude in meters in AMSL altitude, not WGS84 if absolute or relative, above terrain if GLOBAL_TERRAIN_ALT_INT
vx	float	X velocity in NED frame in meter / s
vy	float	Y velocity in NED frame in meter / s
vz	float	Z velocity in NED frame in meter / s
afx	float	X acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afy	float	Y acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
afz	float	Z acceleration or force (if bit 10 of type_mask is set) in NED frame in meter / s^2 or N
yaw	float	yaw setpoint in rad
yaw_rate	float	yaw rate setpoint in rad/s

LOCAL_POSITION_NED_SYSTEM_GLOBAL_OFFSET (#89)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)

x	float	X Position
y	float	Y Position
z	float	Z Position
roll	float	Roll
pitch	float	Pitch
yaw	float	Yaw

HIL_STATE (#90)

DEPRECATED PACKET! Suffers from missing airspeed fields and singularities due to Euler angles. Please use HIL_STATE_QUATERNION instead. Sent from simulation to autopilot. This packet is useful for high throughput applications such as hardware in the loop simulations.

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds since UNIX epoch or microseconds since system boot)
roll	float	Roll angle (rad)
pitch	float	Pitch angle (rad)
yaw	float	Yaw angle (rad)
rollspeed	float	Body frame roll / phi angular speed (rad/s)
pitchspeed	float	Body frame pitch / theta angular speed (rad/s)
yawspeed	float	Body frame yaw / psi angular speed (rad/s)
lat	int32_t	Latitude, expressed as * 1E7
lon	int32_t	Longitude, expressed as * 1E7
alt	int32_t	Altitude in meters, expressed as * 1000 (millimeters)
vx	int16_t	Ground X Speed (Latitude), expressed as m/s * 100
vy	int16_t	Ground Y Speed (Longitude), expressed as m/s * 100
vz	int16_t	Ground Z Speed (Altitude), expressed as m/s * 100
xacc	int16_t	X acceleration (mg)
yacc	int16_t	Y acceleration (mg)
zacc	int16_t	Z acceleration (mg)

HIL_CONTROLS (#91)

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

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds since UNIX epoch or microseconds since system boot)
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	System mode (MAV_MODE)
nav_mode	uint8_t	Navigation mode (MAV_NAV_MODE)

HIL_RC_INPUTS_RAW (#92)

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	Description
time_usec	uint64_t	Timestamp (microseconds since UNIX epoch or microseconds since system boot)
chan1_raw	uint16_t	RC channel 1 value, in microseconds
chan2_raw	uint16_t	RC channel 2 value, in microseconds
chan3_raw	uint16_t	RC channel 3 value, in microseconds
chan4_raw	uint16_t	RC channel 4 value, in microseconds
chan5_raw	uint16_t	RC channel 5 value, in microseconds
chan6_raw	uint16_t	RC channel 6 value, in microseconds
chan7_raw	uint16_t	RC channel 7 value, in microseconds
chan8_raw	uint16_t	RC channel 8 value, in microseconds
chan9_raw	uint16_t	RC channel 9 value, in microseconds
chan10_raw	uint16_t	RC channel 10 value, in microseconds
chan11_raw	uint16_t	RC channel 11 value, in microseconds
chan12_raw	uint16_t	RC channel 12 value, in microseconds
rssi	uint8_t	Receive signal strength indicator, 0: 0%, 255: 100%

OPTICAL_FLOW (#100)

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

Field Name	Type	Description
time_usec	uint64_t	Timestamp (UNIX)
sensor_id	uint8_t	Sensor ID
flow_x	int16_t	Flow in pixels * 10 in x-sensor direction (dezi-pixels)
flow_y	int16_t	Flow in pixels * 10 in y-sensor direction (dezi-pixels)
flow_comp_m_x	float	Flow in meters in x-sensor direction, angular-speed compensated
flow_comp_m_y	float	Flow in meters in y-sensor direction, angular-speed compensated
quality	uint8_t	Optical flow quality / confidence. 0: bad, 255: maximum quality
ground_distance	float	Ground distance in meters. Positive value: distance known. Negative value: Unknown distance

GLOBAL_VISION_POSITION_ESTIMATE (#101)

Field Name	Type	Description
usec	uint64_t	Timestamp (microseconds, synced to UNIX time or since system boot)
x	float	Global X position
y	float	Global Y position
z	float	Global Z position
roll	float	Roll angle in rad
pitch	float	Pitch angle in rad
yaw	float	Yaw angle in rad

VISION_POSITION_ESTIMATE (#102)

Field Name	Type	Description
usec	uint64_t	Timestamp (microseconds, synced to UNIX time or since system boot)
x	float	Global X position
y	float	Global Y position
z	float	Global Z position
roll	float	Roll angle in rad
pitch	float	Pitch angle in rad
yaw	float	Yaw angle in rad

VISION_SPEED_ESTIMATE (#103)

Field Name	Type	Description
usec	uint64_t	Timestamp (microseconds, synced to UNIX time or since system boot)
x	float	Global X speed
y	float	Global Y speed
z	float	Global Z speed

VICON_POSITION_ESTIMATE (#104)

Field Name	Type	Description
usec	uint64_t	Timestamp (microseconds, synced to UNIX time or since system boot)
x	float	Global X position
y	float	Global Y position
z	float	Global Z position
roll	float	Roll angle in rad
pitch	float	Pitch angle in rad
yaw	float	Yaw angle in rad

HIGHRES_IMU (#105)

The IMU readings in SI units in NED body frame

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds, synced to UNIX time or since system boot)
xacc	float	X acceleration (m/s^2)
yacc	float	Y acceleration (m/s^2)
zacc	float	Z acceleration (m/s^2)
xgyro	float	Angular speed around X axis (rad / sec)
ygyro	float	Angular speed around Y axis (rad / sec)

zgyro	float	Angular speed around Z axis (rad / sec)
xmag	float	X Magnetic field (Gauss)
ymag	float	Y Magnetic field (Gauss)
zmag	float	Z Magnetic field (Gauss)
abs_pressure	float	Absolute pressure in millibar
diff_pressure	float	Differential pressure in millibar
pressure_alt	float	Altitude calculated from pressure
temperature	float	Temperature in degrees celsius
fields_updated	uint16_t	Bitmask for fields that have updated since last message, bit 0 = xacc, bit 12: temperature

OPTICAL_FLOW_RAD (#106)

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

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds, synced to UNIX time or since system boot)
sensor_id	uint8_t	Sensor ID
integration_time_us	uint32_t	Integration time in microseconds. Divide integrated_x and integrated_y by the integration time to obtain average flow. The integration time also indicates the.
integrated_x	float	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	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	RH rotation around X axis (rad)
integrated_ygyro	float	RH rotation around Y axis (rad)
integrated_zgyro	float	RH rotation around Z axis (rad)
temperature	int16_t	Temperature * 100 in centi-degrees Celsius
quality	uint8_t	Optical flow quality / confidence. 0: no valid flow, 255: maximum quality
time_delta_distance_us	uint32_t	Time in microseconds since the distance was sampled.
distance	float	Distance to the center of the flow field in meters. Positive value (including zero): distance known. Negative value: Unknown distance.

HIL_SENSOR (#107)

The IMU readings in SI units in NED body frame

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds, synced to UNIX time or since system boot)
xacc	float	X acceleration (m/s^2)
yacc	float	Y acceleration (m/s^2)
zacc	float	Z acceleration (m/s^2)
xgyro	float	Angular speed around X axis in body frame (rad / sec)
ygyro	float	Angular speed around Y axis in body frame (rad / sec)

zgyro	float	Angular speed around Z axis in body frame (rad / sec)
xmag	float	X Magnetic field (Gauss)
ymag	float	Y Magnetic field (Gauss)
zmag	float	Z Magnetic field (Gauss)
abs_pressure	float	Absolute pressure in millibar
diff_pressure	float	Differential pressure (airspeed) in millibar
pressure_alt	float	Altitude calculated from pressure
temperature	float	Temperature in degrees celsius
fields_updated	uint32_t	Bitmask for fields that have updated since last message, bit 0 = xacc, bit 12: temperature, bit 31: full reset of attitude/position/velocities/etc was performed in sim.

SIM_STATE (#108)

Status of simulation environment, if used

Field Name	Type	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	X acceleration m/s/s
yacc	float	Y acceleration m/s/s
zacc	float	Z acceleration m/s/s
xgyro	float	Angular speed around X axis rad/s
ygyro	float	Angular speed around Y axis rad/s
zgyro	float	Angular speed around Z axis rad/s
lat	float	Latitude in degrees
lon	float	Longitude in degrees
alt	float	Altitude in meters
std_dev_horz	float	Horizontal position standard deviation
std_dev_vert	float	Vertical position standard deviation
vn	float	True velocity in m/s in NORTH direction in earth-fixed NED frame
ve	float	True velocity in m/s in EAST direction in earth-fixed NED frame
vd	float	True velocity in m/s in DOWN direction in earth-fixed NED frame

RADIO_STATUS (#109)

Status generated by radio and injected into MAVLink stream.

Field Name	Type	Description
rssi	uint8_t	Local signal strength
remrssi	uint8_t	Remote signal strength

txbuf	uint8_t	Remaining free buffer space in percent.
noise	uint8_t	Background noise level
remnoise	uint8_t	Remote background noise level
rxerrors	uint16_t	Receive errors
fixed	uint16_t	Count of error corrected packets

FILE_TRANSFER_PROTOCOL (#110)

File transfer message

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 entire content of this block is opaque unless you understand any the encoding message_type. The particular encoding used can be extension specific and might not always be documented as part of the mavlink specification.

TIMESYNC (#111)

Time synchronization message.

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

CAMERA_TRIGGER (#112)

Camera-IMU triggering and synchronisation message.

Field Name	Type	Description
time_usec	uint64_t	Timestamp for the image frame in microseconds
seq	uint32_t	Image frame sequence

HIL_GPS (#113)

The global position, as returned by the Global Positioning System (GPS). This is NOT the global position estimate of the sytem, but rather a RAW sensor value. See message GLOBAL_POSITION for the global position estimate. Coordinate frame is right-handed, Z-axis up (GPS frame).

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds since UNIX epoch or microseconds

		since system boot)
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	Latitude (WGS84), in degrees * 1E7
lon	int32_t	Longitude (WGS84), in degrees * 1E7
alt	int32_t	Altitude (AMSL, not WGS84), in meters * 1000 (positive for up)
eph	uint16_t	GPS HDOP horizontal dilution of position in cm (m*100). If unknown, set to: 65535
epv	uint16_t	GPS VDOP vertical dilution of position in cm (m*100). If unknown, set to: 65535
vel	uint16_t	GPS ground speed (m/s * 100). If unknown, set to: 65535
vn	int16_t	GPS velocity in cm/s in NORTH direction in earth-fixed NED frame
ve	int16_t	GPS velocity in cm/s in EAST direction in earth-fixed NED frame
vd	int16_t	GPS velocity in cm/s in DOWN direction in earth-fixed NED frame
cog	uint16_t	Course over ground (NOT heading, but direction of movement) in degrees * 100, 0.0..359.99 degrees. If unknown, set to: 65535
satellites_visible	uint8_t	Number of satellites visible. If unknown, set to 255

HIL_OPTICAL_FLOW (#114)

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

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds, synced to UNIX time or since system boot)
sensor_id	uint8_t	Sensor ID
integration_time_us	uint32_t	Integration time in microseconds. Divide integrated_x and integrated_y by the integration time to obtain average flow. The integration time also indicates the.
integrated_x	float	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	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	RH rotation around X axis (rad)
integrated_ygyro	float	RH rotation around Y axis (rad)
integrated_zgyro	float	RH rotation around Z axis (rad)
temperature	int16_t	Temperature * 100 in centi-degrees Celsius
quality	uint8_t	Optical flow quality / confidence. 0: no valid flow, 255: maximum quality
time_delta_distance_us	uint32_t	Time in microseconds since the distance was sampled.
distance	float	Distance to the center of the flow field in meters. Positive value (including zero): distance known. Negative value: Unknown distance.

HIL_STATE_QUATERNION (#115)

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	Description
time_usec	uint64_t	Timestamp (microseconds since UNIX epoch or microseconds since system boot)
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	Body frame roll / phi angular speed (rad/s)
pitchspeed	float	Body frame pitch / theta angular speed (rad/s)
yawspeed	float	Body frame yaw / psi angular speed (rad/s)
lat	int32_t	Latitude, expressed as * 1E7
lon	int32_t	Longitude, expressed as * 1E7
alt	int32_t	Altitude in meters, expressed as * 1000 (millimeters)
vx	int16_t	Ground X Speed (Latitude), expressed as m/s * 100
vy	int16_t	Ground Y Speed (Longitude), expressed as m/s * 100
vz	int16_t	Ground Z Speed (Altitude), expressed as m/s * 100
ind_airspeed	uint16_t	Indicated airspeed, expressed as m/s * 100
true_airspeed	uint16_t	True airspeed, expressed as m/s * 100
xacc	int16_t	X acceleration (mg)
yacc	int16_t	Y acceleration (mg)
zacc	int16_t	Z acceleration (mg)

SCALED_IMU2 (#116)

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

Field Name	Type	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
xacc	int16_t	X acceleration (mg)
yacc	int16_t	Y acceleration (mg)
zacc	int16_t	Z acceleration (mg)
xgyro	int16_t	Angular speed around X axis (millirad /sec)
ygyro	int16_t	Angular speed around Y axis (millirad /sec)
zgyro	int16_t	Angular speed around Z axis (millirad /sec)
xmag	int16_t	X Magnetic field (milli tesla)
ymag	int16_t	Y Magnetic field (milli tesla)
zmag	int16_t	Z Magnetic field (milli tesla)

LOG_REQUEST_LIST (#117)

Request a list of available logs. On some systems calling this may stop on-board logging until LOG_REQUEST_END is called.

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)

Reply to LOG_REQUEST_LIST

Field Name	Type	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	UTC timestamp of log in seconds since 1970, or 0 if not available
size	uint32_t	Size of the log (may be approximate) in bytes

LOG_REQUEST_DATA (#119)

Request a chunk of a log

Field Name	Type	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	Number of bytes

LOG_DATA (#120)

Reply to LOG_REQUEST_DATA

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

LOG_ERASE (#121)

Erase all logs

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

LOG_REQUEST_END (#122)

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)

data for injecting into the onboard GPS (used for DGPS)

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

GPS2_RAW (#124)

Second GPS data. Coordinate frame is right-handed, Z-axis up (GPS frame).

Field Name	Type	Description
time_usec	uint64_t	Timestamp (microseconds since UNIX epoch or microseconds since system boot)
fix_type	uint8_t	0-1: no fix, 2: 2D fix, 3: 3D fix, 4: DGPS fix, 5: RTK 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	Latitude (WGS84), in degrees * 1E7
lon	int32_t	Longitude (WGS84), in degrees * 1E7
alt	int32_t	Altitude (AMSL, not WGS84), in meters * 1000 (positive for up)
eph	uint16_t	GPS HDOP horizontal dilution of position in cm (m*100). If unknown, set to: UINT16_MAX
epv	uint16_t	GPS VDOP vertical dilution of position in cm (m*100). If unknown, set to: UINT16_MAX
vel	uint16_t	GPS ground speed (m/s * 100). If unknown, set to: UINT16_MAX
cog	uint16_t	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 255
dgps_numch	uint8_t	Number of DGPS satellites
dgps_age	uint32_t	Age of DGPS info

POWER_STATUS (#125)

Power supply status

Field Name	Type	Description
Vcc	uint16_t	5V rail voltage in millivolts
Vservo	uint16_t	servo rail voltage in millivolts
flags	uint16_t	power supply status flags (see MAV_POWER_STATUS enum)

SERIAL_CONTROL (#126)

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	Description
device	uint8_t	See SERIAL_CONTROL_DEV enum
flags	uint8_t	See SERIAL_CONTROL_FLAG enum
timeout	uint16_t	Timeout for reply data in milliseconds
baudrate	uint32_t	Baudrate of transfer. Zero means no change.
count	uint8_t	how many bytes in this transfer
data	uint8_t[70]	serial data

GPS_RTK (#127)

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

Field Name	Type	Description
time_last_baseline_ms	uint32_t	Time since boot of last baseline message received in ms.
rtk_receiver_id	uint8_t	Identification of connected RTK receiver.
wn	uint16_t	GPS Week Number of last baseline
tow	uint32_t	GPS Time of Week of last baseline
rtk_health	uint8_t	GPS-specific health report for RTK data.
rtk_rate	uint8_t	Rate of baseline messages being received by GPS, in HZ
nsats	uint8_t	Current number of sats used for RTK calculation.
baseline_coords_type	uint8_t	Coordinate system of baseline. 0 == ECEF, 1 == NED
baseline_a_mm	int32_t	Current baseline in ECEF x or NED north component in mm.
baseline_b_mm	int32_t	Current baseline in ECEF y or NED east component in mm.
baseline_c_mm	int32_t	Current baseline in ECEF z or NED down component in mm.
accuracy	uint32_t	Current estimate of baseline accuracy.
iar_num_hypotheses	int32_t	Current number of integer ambiguity hypotheses.

GPS2_RTK (#128)

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

Field Name	Type	Description
time_last_baseline_ms	uint32_t	Time since boot of last baseline message received in ms.
rtk_receiver_id	uint8_t	Identification of connected RTK receiver.
wn	uint16_t	GPS Week Number of last baseline
tow	uint32_t	GPS Time of Week of last baseline
rtk_health	uint8_t	GPS-specific health report for RTK data.
rtk_rate	uint8_t	Rate of baseline messages being received by GPS, in HZ
nsats	uint8_t	Current number of sats used for RTK calculation.
baseline_coords_type	uint8_t	Coordinate system of baseline. 0 == ECEF, 1 == NED
baseline_a_mm	int32_t	Current baseline in ECEF x or NED north component in mm.
baseline_b_mm	int32_t	Current baseline in ECEF y or NED east component in mm.
baseline_c_mm	int32_t	Current baseline in ECEF z or NED down component in mm.

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

SCALED_IMU3 (#129)

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

Field Name	Type	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
xacc	int16_t	X acceleration (mg)
yacc	int16_t	Y acceleration (mg)
zacc	int16_t	Z acceleration (mg)
xgyro	int16_t	Angular speed around X axis (millirad /sec)
ygyro	int16_t	Angular speed around Y axis (millirad /sec)
zgyro	int16_t	Angular speed around Z axis (millirad /sec)
xmag	int16_t	X Magnetic field (milli tesla)
ymag	int16_t	Y Magnetic field (milli tesla)
zmag	int16_t	Z Magnetic field (milli tesla)

DATA_TRANSMISSION_HANDSHAKE (#130)

Field Name	Type	Description
type	uint8_t	type of requested/acknowledged data (as defined in ENUM DATA_TYPES in mavlink/include/mavlink_types.h)
size	uint32_t	total data size in bytes (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 beeing sent (set on ACK only)
payload	uint8_t	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 out of [1,100]

ENCAPSULATED_DATA (#131)

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)

Field Name	Type	Description
time_boot_ms	uint32_t	Time since system boot

min_distance	uint16_t	Minimum distance the sensor can measure in centimeters
max_distance	uint16_t	Maximum distance the sensor can measure in centimeters
current_distance	uint16_t	Current distance reading
type	uint8_t	Type from MAV_DISTANCE_SENSOR enum.
id	uint8_t	Onboard ID of the sensor
orientation	uint8_t	Direction the sensor faces from MAV_SENSOR_ORIENTATION enum.
covariance	uint8_t	Measurement covariance in centimeters, 0 for unknown / invalid readings

TERRAIN_REQUEST (#133)

Request for terrain data and terrain status

Field Name	Type	Description
lat	int32_t	Latitude of SW corner of first grid (degrees *10^7)
lon	int32_t	Longitude of SW corner of first grid (in degrees *10^7)
grid_spacing	uint16_t	Grid spacing in meters
mask	uint64_t	Bitmask of requested 4x4 grids (row major 8x7 array of grids, 56 bits)

TERRAIN_DATA (#134)

Terrain data sent from GCS. The lat/lon and grid_spacing must be the same as a lat/lon from a TERRAIN_REQUEST

Field Name	Type	Description
lat	int32_t	Latitude of SW corner of first grid (degrees *10^7)
lon	int32_t	Longitude of SW corner of first grid (in degrees *10^7)
grid_spacing	uint16_t	Grid spacing in meters
gridbit	uint8_t	bit within the terrain request mask
data	int16_t[16]	Terrain data in meters AMSL

TERRAIN_CHECK (#135)

Request that the vehicle report terrain height at the given location. Used by GCS to check if vehicle has all terrain data needed for a mission.

Field Name	Type	Description
lat	int32_t	Latitude (degrees *10^7)
lon	int32_t	Longitude (degrees *10^7)

TERRAIN_REPORT (#136)

Response from a TERRAIN_CHECK request

Field Name	Type	Description
lat	int32_t	Latitude (degrees *10^7)
lon	int32_t	Longitude (degrees *10^7)
spacing	uint16_t	grid spacing (zero if terrain at this location unavailable)
terrain_height	float	Terrain height in meters AMSL
current_height	float	Current vehicle height above lat/lon terrain height (meters)
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)

Barometer readings for 2nd barometer

Field Name	Type	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
press_abs	float	Absolute pressure (hectopascal)
press_diff	float	Differential pressure 1 (hectopascal)
temperature	int16_t	Temperature measurement (0.01 degrees celsius)

ATT_POS_MOCAP (#138)

Motion capture attitude and position

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
q	float[4]	Attitude quaternion (w, x, y, z order, zero-rotation is 1, 0, 0, 0)
x	float	X position in meters (NED)
y	float	Y position in meters (NED)
z	float	Z position in meters (NED)

SET_ACTUATOR_CONTROL_TARGET (#139)

Set the vehicle attitude and body angular rates.

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
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)

Set the vehicle attitude and body angular rates.

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
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)

The current system altitude.

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
altitude_monotonic	float	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	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 AMSL by default and not the WGS84 altitude.
altitude_local	float	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	This is the altitude above the home position. It resets on each change of the current home position.
altitude_terrain	float	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	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)

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)

Barometer readings for 3rd barometer

Field Name	Type	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
press_abs	float	Absolute pressure (hectopascal)
press_diff	float	Differential pressure 1 (hectopascal)
temperature	int16_t	Temperature measurement (0.01 degrees celsius)

FOLLOW_TARGET (#144)

current motion information from a designated system

Field Name	Type	Description
timestamp	uint64_t	Timestamp in milliseconds 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	Latitude (WGS84), in degrees * 1E7
lon	int32_t	Longitude (WGS84), in degrees * 1E7
alt	float	AMSL, in meters
vel	float[3]	target velocity (0,0,0) for unknown
acc	float[3]	linear target acceleration (0,0,0) for unknown
attitude_q	float[4]	(1 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)

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

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
x_acc	float	X acceleration in body frame
y_acc	float	Y acceleration in body frame
z_acc	float	Z acceleration in body frame
x_vel	float	X velocity in body frame
y_vel	float	Y velocity in body frame
z_vel	float	Z velocity in body frame
x_pos	float	X position in local frame
y_pos	float	Y position in local frame
z_pos	float	Z position in local frame
airspeed	float	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	Angular rate in roll axis
pitch_rate	float	Angular rate in pitch axis
yaw_rate	float	Angular rate in yaw axis

BATTERY_STATUS (#147)

Battery information

Field Name	Type	Description
id	uint8_t	Battery ID
battery_function	uint8_t	Function of the battery
type	uint8_t	Type (chemistry) of the battery
temperature	int16_t	Temperature of the battery in centi-degrees celsius. INT16_MAX for unknown temperature.
voltages	uint16_t[10]	Battery voltage of cells, in millivolts (1 = 1 millivolt). Cells above the valid cell count for this battery should have the UINT16_MAX value.
current_battery	int16_t	Battery current, in 10*milliamperes (1 = 10 milliampere), -1: autopilot does not measure the current
current_consumed	int32_t	Consumed charge, in milliampere hours (1 = 1 mAh), -1: autopilot does not provide mAh consumption estimate
energy_consumed	int32_t	Consumed energy, in 100*Joules (intergrated U*I*dt) (1 = 100 Joule), -1: autopilot does not provide energy consumption estimate
battery_remaining	int8_t	Remaining battery energy: (0%: 0, 100%: 100), -1: autopilot does not estimate the remaining battery

AUTOPILOT_VERSION (#148)

Version and capability of autopilot software

Field Name	Type	Description
capabilities	uint64_t	bitmask of capabilities (see MAV_PROTOCOL_CAPABILITY

		enum)
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 bytes should be silicon ID, if any)
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

LANDING_TARGET (#149)

The location of a landing area captured from a downward facing camera

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
target_num	uint8_t	The ID of the target if multiple targets are present
frame	uint8_t	MAV_FRAME enum specifying the whether the following feilds are earth-frame, body-frame, etc.
angle_x	float	X-axis angular offset (in radians) of the target from the center of the image
angle_y	float	Y-axis angular offset (in radians) of the target from the center of the image
distance	float	Distance to the target from the vehicle in meters
size_x	float	Size in radians of target along x-axis
size_y	float	Size in radians of target along y-axis

ESTIMATOR_STATUS (#230)

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 innovaton test ratios show the magnitude of the sensor innovation divided by the innovation check threshold. Under normal operation the innovaton 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	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
flags	uint16_t	Integer bitmask indicating which EKF outputs are valid. See

		definition for ESTIMATOR_STATUS_FLAGS.
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	Horizontal position 1-STD accuracy relative to the EKF local origin (m)
pos_vert_accuracy	float	Vertical position 1-STD accuracy relative to the EKF local origin (m)

WIND_COV (#231)

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
wind_x	float	Wind in X (NED) direction in m/s
wind_y	float	Wind in Y (NED) direction in m/s
wind_z	float	Wind in Z (NED) direction in m/s
var_horiz	float	Variability of the wind in XY. RMS of a 1 Hz lowpassed wind estimate.
var_vert	float	Variability of the wind in Z. RMS of a 1 Hz lowpassed wind estimate.
wind_alt	float	AMSL altitude (m) this measurement was taken at
horiz_accuracy	float	Horizontal speed 1-STD accuracy
vert_accuracy	float	Vertical speed 1-STD accuracy

GPS_INPUT (#232)

GPS sensor input message. This is a raw sensor value sent by the GPS. This is NOT the global position estimate of the sytem.

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
gps_id	uint8_t	ID of the GPS for multiple GPS inputs
ignore_flags	uint16_t	Flags indicating which fields to ignore (see GPS_INPUT_IGNORE_FLAGS enum). All other fields must be provided.
time_week_ms	uint32_t	GPS time (milliseconds 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	Latitude (WGS84), in degrees * 1E7
lon	int32_t	Longitude (WGS84), in degrees * 1E7
alt	float	Altitude (AMSL, not WGS84), in m (positive for up)
hdop	float	GPS HDOP horizontal dilution of position in m
vdop	float	GPS VDOP vertical dilution of position in m
vn	float	GPS velocity in m/s in NORTH direction in earth-fixed NED frame
ve	float	GPS velocity in m/s in EAST direction in earth-fixed NED frame
vd	float	GPS velocity in m/s in DOWN direction in earth-fixed NED frame

speed_accuracy	float	GPS speed accuracy in m/s
horiz_accuracy	float	GPS horizontal accuracy in m
vert_accuracy	float	GPS vertical accuracy in m
satellites_visible	uint8_t	Number of satellites visible.

GPS_RTCM_DATA (#233)

WORK IN PROGRESS! RTCM message for injecting into the onboard GPS (used for DGPS)

Field Name	Type	Description
flags	uint8_t	LSB: 1 means message is fragmented
len	uint8_t	data length
data	uint8_t[180]	RTCM message (may be fragmented)

LANDING_MAP (#240)

Quality data about specific landing positions

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
best_x	float	Best landing position on X-axis
best_y	float	Best landing position on Y-axis
best_z	float	Best landing position on Z-axis
local_x	float	Position on X-axis
local_y	float	Position on Y-axis
local_z	float	Position on Z-axis
factors	uint8_t[169]	LSB 0-2: Score between 0 and 7, LSB 3-7: Distance to the vehicle in meters.

VIBRATION (#241)

Vibration levels and accelerometer clipping

Field Name	Type	Description
time_usec	uint64_t	Timestamp (micros since boot or Unix epoch)
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)

This message can be requested by sending the MAV_CMD_GET_HOME_POSITION command. The position the system will return to and land on. The position is set automatically by the system during the takeoff in case it was not explicitely set by the operator before or after. The position the system will return to and land on. 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.

Field Name	Type	Description
latitude	int32_t	Latitude (WGS84), in degrees * 1E7
longitude	int32_t	Longitude (WGS84, in degrees * 1E7
altitude	int32_t	Altitude (AMSL), in meters * 1000 (positive for up)
x	float	Local X position of this position in the local coordinate frame
y	float	Local Y position of this position in the local coordinate frame
z	float	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	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	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	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.

SET_HOME_POSITION (#243)

The position the system will return to and land on. The position is set automatically by the system during the takeoff in case it was not explicitely set by the operator before or after. 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.

Field Name	Type	Description
target_system	uint8_t	System ID.
latitude	int32_t	Latitude (WGS84), in degrees * 1E7
longitude	int32_t	Longitude (WGS84, in degrees * 1E7
altitude	int32_t	Altitude (AMSL), in meters * 1000 (positive for up)
x	float	Local X position of this position in the local coordinate frame
y	float	Local Y position of this position in the local coordinate frame
z	float	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	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	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	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.

MESSAGE_INTERVAL (#244)

This interface replaces DATA_STREAM

Field Name	Type	Description
message_id	uint16_t	The ID of the requested MAVLink message. v1.0 is limited to 254 messages.
interval_us	int32_t	The interval between two messages, in microseconds. 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)

Provides state for additional features

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

ADSB_VEHICLE (#246)

The location and information of an ADSB vehicle

Field Name	Type	Description
ICAO_address	uint32_t	ICAO address
lat	int32_t	Latitude, expressed as degrees * 1E7
lon	int32_t	Longitude, expressed as degrees * 1E7
altitude_type	uint8_t	Type from ADSB_ALTITUDE_TYPE enum

altitude	int32_t	Altitude(ASL) in millimeters
heading	uint16_t	Course over ground in centidegrees
hor_velocity	uint16_t	The horizontal velocity in centimeters/second
ver_velocity	int16_t	The vertical velocity in centimeters/second, positive is up
callsign	char[9]	The callsign, 8+null
emitter_type	uint8_t	Type from ADSB_EMITTER_TYPE enum
tslc	uint8_t	Time since last communication in seconds
flags	uint16_t	Flags to indicate various statuses including valid data fields
squawk	uint16_t	Squawk code

COLLISION (#247)

Information about a potential collision

Field Name	Type	Description
src	uint8_t	Collision data source
id	uint32_t	Unique identifier, domain based on src field
action	uint8_t	Action that is being taken to avoid this collision
threat_level	uint8_t	How concerned the aircraft is about this collision
time_to_minimum_delta	float	Estimated time until collision occurs (seconds)
altitude_minimum_delta	float	Closest vertical distance in meters between vehicle and object
horizontal_minimum_delta	float	Closest horizontal distance in meteres between vehicle and object

V2_EXTENSION (#248)

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/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 is defined by the remaining message length when subtracting the header and other fields. The entire content of this block is opaque unless you understand any 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)

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)

Field Name	Type	Description
name	char[10]	Name
time_usec	uint64_t	Timestamp
x	float	x
y	float	y
z	float	z

NAMED_VALUE_FLOAT (#251)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
name	char[10]	Name of the debug variable
value	float	Floating point value

NAMED_VALUE_INT (#252)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
name	char[10]	Name of the debug variable
value	int32_t	Signed integer value

STATUSTEXT (#253)

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	Description
severity	uint8_t	Severity of status. Relies on the definitions within RFC-5424. See enum MAV_SEVERITY.
text	char[50]	Status text message, without null termination character

DEBUG (#254)

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	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
ind	uint8_t	index of debug variable
value	float	DEBUG value

SETUP_SIGNING (#256)

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)

Report button state change

Field Name	Type	Description
time_boot_ms	uint32_t	Timestamp (milliseconds since system boot)
last_change_ms	uint32_t	Time of last change of button state
state	uint8_t	Bitmap state of buttons

PLAY_TUNE (#258)

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

Messages are defined by the common.xml file. The C packing/unpacking code is generated from this specification, as well as the HTML documentaiton in the section above.

The XML displayed here is updated on every commit and therefore up-to-date.

mavlink